

NOAA TECHNICAL MEMORANDUM NMFS SEFC-151



FISHERIES LIBRARY
76 VIRGINIA BEACH DR.
MIAMI, FL 33149

GRAPHICS -- AN ANTHOLOGY OF PROGRAMS

Lysander Ng

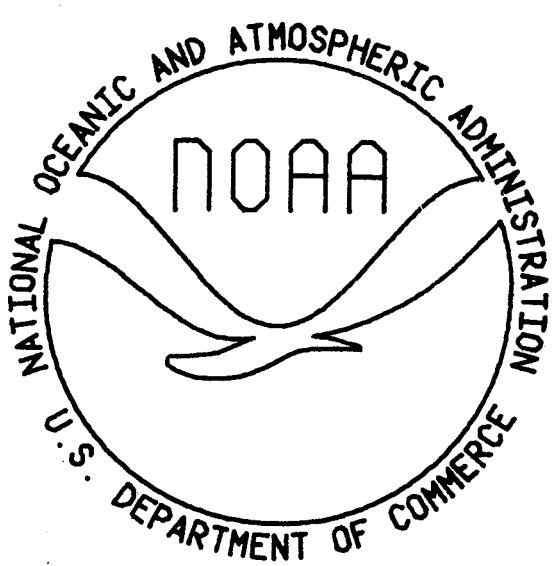
November 1984

U. S. DEPARTMENT OF COMMERCE
National Oceanic and Atmospheric Administration
National Marine Fisheries Service
Southeast Fisheries Center
Charleston Laboratory
P. O. Box 12607
Charleston, South Carolina 29412 - 0607

NOAA TECHNICAL MEMORANDUM NMFS SEFC-151

Technical Memorandums are used for documentation and timely communication of preliminary results, interim reports, or special-purpose information, and have not received complete formal review, editorial control, or detailed editing.

GRAPHICS -- AN ANTHOLOGY OF PROGRAMS



Lysander Ng

November 1984

U. S. DEPARTMENT OF COMMERCE

Malcolm Baldrige, Secretary

NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION

Anthony J. Calio, Acting Administrator

NATIONAL MARINE FISHERIES SERVICE

William G. Gordon, Assistant Administrator for Fisheries

ABSTRACT

Eight graphics programs useful for presenting and analyzing data are operational on the Tektronix^{1/} plotters and 4050 series desktop processors. These programs are entitled 1) "Pie-chart," 2) "3-d Transformation," 3) "Bargraph," 4) "Fractional Bargraph," 5) "Legend Maker," 6) "X-Y Graph," 7) "Text Plot," and 8) "Block Plot."

The capabilities, examples, and program listing of each program are presented.

1/ The National Marine Fisheries Service (NMFS) does not necessarily approve, recommend, or endorse any proprietary product or proprietary material mentioned in this publication. No reference shall be made to NMFS, or this publication furnished by NMFS, in any advertising or sales promotion which would indicate or imply that NMFS approves, recommends, or endorses any proprietary products or proprietary material herein, or which has as its purpose an intent to cause directly or indirectly the advertised product to be used or purchased because of this NMFS publication.

TABLE OF CONTENTS

	Page
1. INTRODUCTION.....	1
2. PIE-CHART.....	3
Output.....	3
Input Variables.....	5
Pie Locations and Sizes.....	5
Labels.....	9
Data to be Represented.....	9
Shades.....	9
3. 3-D TRANSFORMATION.....	15
Scaling.....	16
Rotation.....	16
Data Input.....	16
Hidden lines Removal.....	25
4. BARGRAPH.....	31
Input Variables.....	31
Labels.....	36
Enhancements.....	36
5. FRACTIONAL BARGRAPH.....	39
Input Variables.....	39
Enhancements.....	41
6. LEGEND MAKER.....	43
Enhancements and Captions.....	43
Size and Location.....	43
Input Variables.....	45

TABLE OF CONTENTS (Continued)

	Page
7. X-Y GRAPH.....	47
Original Capabilities.....	47
Added Capabilities.....	47
Dot and Dash Modes.....	50
8. TEXT PLOT.....	53
Data Entry.....	53
Other UDK Functions.....	55
9. BLOCK PLOT.....	59
Input Variables.....	59
Thickness of Lines.....	59
Character Size.....	59
APPENDICES	
A. Pie-chart Program Listing.....	65
B. 3-d Transformation Program Listing.....	73
C. Bargraph Program Listing.....	87
D. Fractional Bargraph Program Listing.....	93
E. Legend Maker Program Listing.....	99
F. X-Y Graph Program Listing.....	103
G. Text Plot Program Listing.....	115
H. Block Plot Program Listing.....	125

LIST OF TABLES

	Page
2.1. Input variables for "Pie-chart".....	6
2.2. Relationships between shades and distances between dots.....	10
4.1. Input variables for "Bargraph".....	35
4.2. Bargraph enhancements.....	37
5.1. Fractional bargraph enhancements.....	42
6.1. Input variables for "Legend Maker".....	46
8.1. Functions of the reserved character '&'.....	54
9.1. Input variables for "Block Plot".....	60

LIST OF FIGURES

	Page
2.1. Example of outputs during "Pie-chart" interactive sessions.....	4
2.2. Example of pie-charts with various pie sizes.....	7
2.3. Example of pie-charts with all of the five shades.....	8
2.4. Example of pie-charts with a staggered pattern of shades.....	12
3.1. Demonstration of the scaling transformation.....	17
3.2. Demonstration of the counter-clockwise rotation function.....	18
3.3. Collection of figures after rotations.....	19
3.4. Construction example of 3-d figures.....	20
3.5. Example of 3-d surfaces.....	22
3.6. 3-d surface of Figure 3.5 with a different perspective.....	23
3.7. Demonstration of construction of a sphere.....	24
3.8. Demonstration of the hidden lines removal option.....	26
3.9. Model for explaining the hidden line algorithm.....	28
3.10. Decision tree of the hidden line algorithm.....	29
4.1. Example of bargraphs with various numbers of discrete variables.	32
4.2. Example of bargraphs with a rotated y axis label.....	33
4.3. Example of bargraphs with all of the eight enhancements.....	34
5.1. Example of fractional bargraphs.....	40
6.1. Example of legends.....	44
7.1. Example of X-Y graphs with all of the four plot modes.....	48
7.2. Example of X-Y graphs with the linear regression option.....	49
8.1. Example of stored text of "Text Plot".....	56
8.2. Example of final "Text Plot" outputs.....	57
9.1. Demonstration of the mechanism for producing block letters.....	61
9.2. Example of block plots.....	62

1. INTRODUCTION

"A picture is worth a thousand words." The purpose of this report is to supply researchers with the means to use a form of a universal pictorial language -- graphical representation of data. Possessing the qualities of pictorial models, graphs are widely adopted, especially within scientific circles. Being data analytical tools, graphs are not only communicative, but they can also assist us in deriving insights. A well chosen graph can represent voluminous amounts of data, can ease committing of concepts to memory, can describe salient data structure, and can aid in the drawing of analytical inferences.

Many techniques have been developed to graphically represent data, and computer graphics is at the leading edge of this technology. Various graphical forms are widely used and preferred by the scientific communities.

This report is intended to introduce researchers to eight user-friendly graphical routines which are operational on the Tektronix plotters and 4050 series desktops. This report allows researchers a quick reference to some of the available graphical capabilities. Six of the programs presented here, 1) "Pie-chart," 2) "3-d Transformation," 3) "Bargraph," 4) "Fractional Bar-graph," 5) "Legend Maker," and 6) "X-Y Graph," can provide one with a substantial range of graphical applications. Also included is the "Text Plot" program which allows one to plot text materials, especially those involving Greek alphabets and mathematical symbols. Finally, the "Block Plot" program is useful for plotting block letters for visual displays.

This guide has nine sections. Each of the Sections, 2 through 9, corresponds to a program. The listings of the programs are provided in the Appendices.

All eight programs are interactive. Data and parameters are supplied by users through question-answer sessions. Although users do not need to have knowledge in programming languages in order to take advantage of the programs, they are assumed to have operational knowledge of the graphical terminal and the plotter.

Both the terminal screen and the plotter may receive outputs from all of the programs except for the "Block Plot" program. Outputs of the "Block Plot" program may only be sent to the plotter. To specify the output location, user enter the parameters during interactive sessions. Outputs can generally be completed more quickly on the screen than on the plotter. The graphs may be viewed first on the screen, and then when satisfied, they may be sent to the plotter.

Except for the "X-Y Graph" routine, all the programs represent original contributions by the author. The "X-Y Graph" program is supplied by the manufacturer and modified by the author to include additional capabilities.

Programming techniques, which enable certain plotting capabilities, are presented for the technically oriented users. Readers are encouraged to contact the author in dealing with questions on aspects of the programs, includir their capabilities and constructions. Suggestions, which may improve the performances of the programs, are also welcome.

2. PIE-CHART

Several pies of various sizes can be plotted in one output. Each portion of the pies can be shaded for contrast. Each graph, pie, and portion can be labelled with various character sizes.

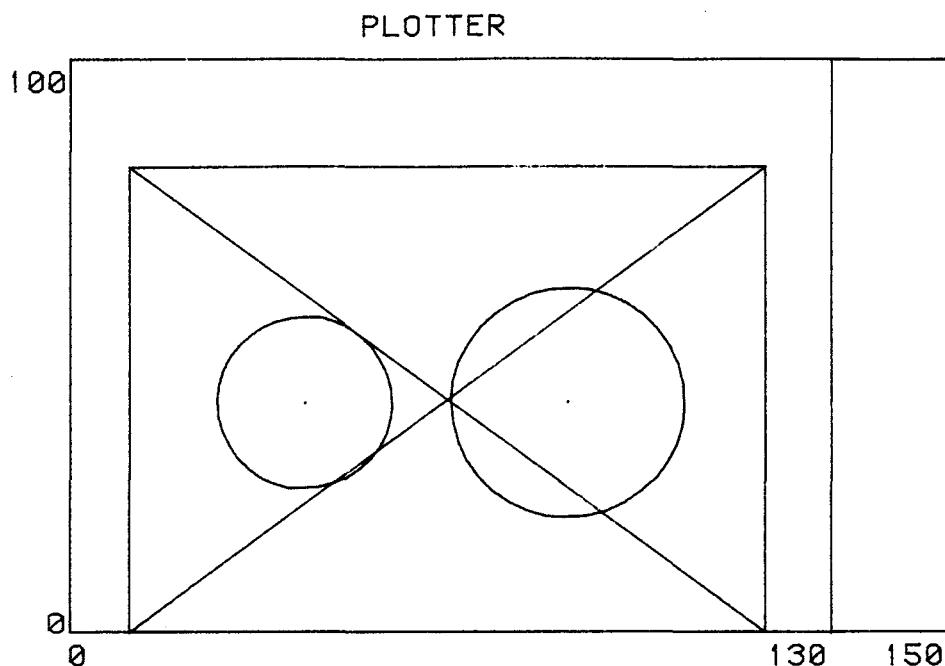
Output

In order to obtain perfectly rounded circles for the pies, it is best to retain the default horizontal and vertical margins of the plotter, and not to reset them. Otherwise, the pies may appear as ellipses.

To ensure that the graph is placed on the user-desired area of the plotter, the user must identify the location of the paper for the processor during the interactive session. After the processor has this user-input information, the location of the paper relative to the plotter is presented on the terminal screen. Figure 2.1 minus the circles is an example. The circles are included, such as in Figure 2.1, after the locations and sizes of pies are assigned. The inner rectangle with an "X" connecting the opposite corners represents the paper. The outer rectangles represent the plotter and the terminal screen surfaces, with the plotter being slightly wider.

Points on the plotter and screen surfaces are addressable by indicating the x and y coordinates in that order. The x coordinate ranges from 0 to 130 from left to right on the terminal screen, while from 0 to 150 on the plotter surface. The plotter has an additional 20 units to the right as compared to the terminal screen. The y coordinate ranges from 0 to 100 from bottom to top for both the plotter and the screen.

If the plotter is not selected as an output medium, preparations for plotter outputs such as margin determination of the paper would be skipped. In



ENCLOSING RECTANGLE IS THE OUTLINE OF PLOTTER SURFACE.
CROSSED AREA IS YOUR PAPER.

THE CIRCLES INDICATE POSITIONS OF PIE CHART.
DO YOU WANT TO MAKE ANY CHANGES ON THE POSITION? (N,Y)

Figure 2.1. Example of an output on the terminal screen during the interactive session. This plot depicts the location of a piece of paper (the crossed rectangle) in relation to the plotter surface (the outermost rectangle). The screen surface is slightly smaller than the plotter surface, lacking the 20 units at the right margin. The circles indicate the positions of pies. The user can make modifications, such as assigning new locations to the pies, by working interactively with the system.

order to reclaim the plotter output option, the user may select item 4 of the prompt available after the graph is plotted on the screen.

Input Variables

The user-inputs prompted by the program are identified by variable names. Table 2.1 lists the user-input variables and their names respective to the sequence of program prompts.

Pie Locations and Sizes

Figures 2.2 and 2.3 are two examples of the final output products.

The pies may be assigned to anywhere on the plotting surface according to user-specifications. The pie locations are dependent on their x and y coordinates. (For additional explanation of proper coordinates, refer to the "Plotter Output" subsection.) The placements of paper and the designations of coordinates should be coordinated in order to produce desirable outputs, i.e., for instance, with plotted figures centered on paper. As an example of input locations, in Figure 2.1, the (x,y) coordinates are (40,40) for the small circle center, and (85,40) for the large circle center.

More than one size of pies may be produced in the same output. To produce Figure 2.2, for example, three different sizes were requested. In Figure 2.2, the radii are 8, 10, and 12 units in length for the three different pies.

After each input of the center coordinates and radii, the program displays on the screen a plot similar to that of Figure 2.1, giving the locations of pies relative to the plotting surface. If the user is not satisfied with the locations or sizes of the pies, the user has the options to make alterations interactively until requirements are met.

Table 2.1. List of input variables: programmer-assigned variable names and their corresponding descriptions.

<u>Programmer-assigned Name</u>	<u>Variable Description</u>
T	Number of pies to be plotted
C	x coordinates of centers of pies
D	y coordinates of centers of pies
B	Radii of pies
P	Number of portions in each pie
M	Shade number for each portion of pies
A	Percentages of portions of pies
S\$	Graph label
Z1	Width of characters for graph label
Z2	Height of characters for graph label
R\$	Pie labels
Z3	Width of characters for pie labels
Z4	Height of characters for pie labels
Q\$	Portion labels
Z5	Width of characters for portion labels
Z6	Height of characters for portion labels
O	Output device number

PROXIMATE COMPOSITION

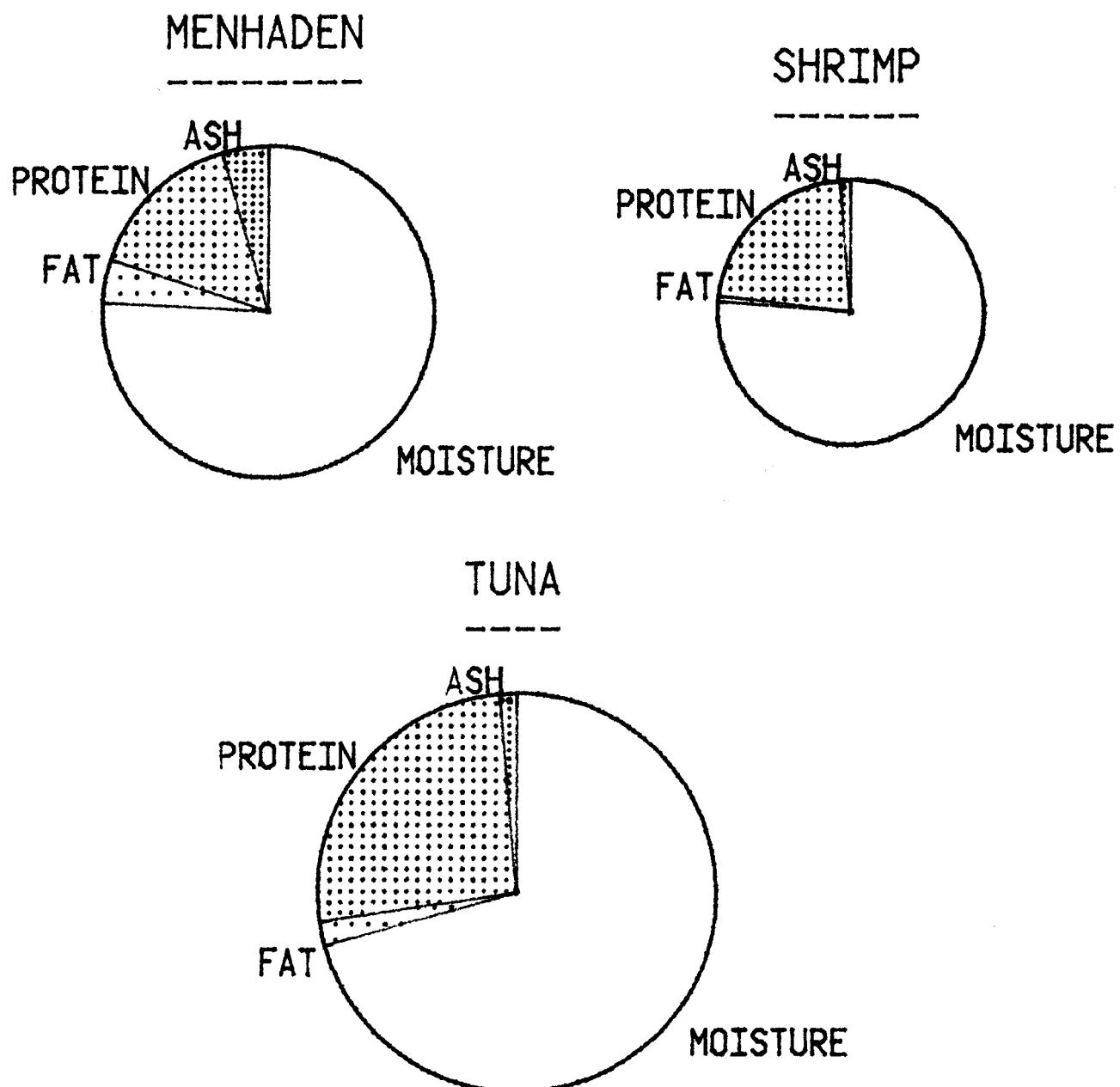
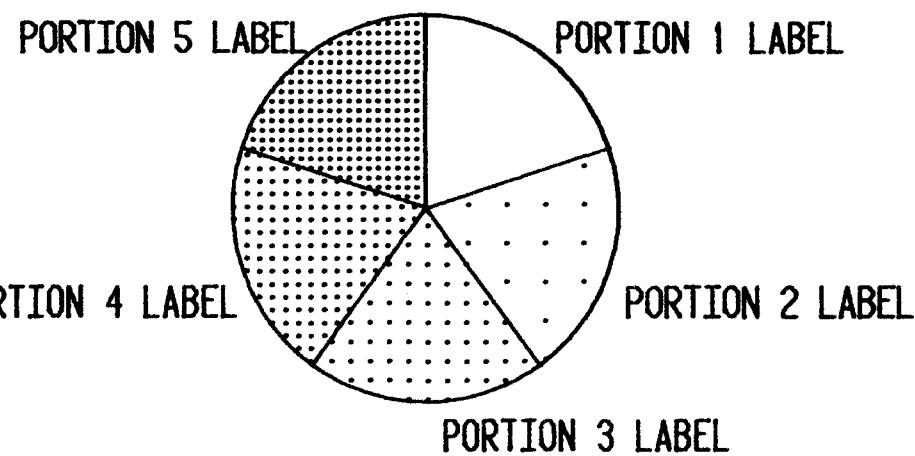


Figure 2.2. Example of an end product. Pie size is one of the mandatory user-inputs. More than one size is possible on the same plot by assigning various radii to different pie-circles.

GRAPH LABEL

PIE #1 LABEL



PIE #2 LABEL

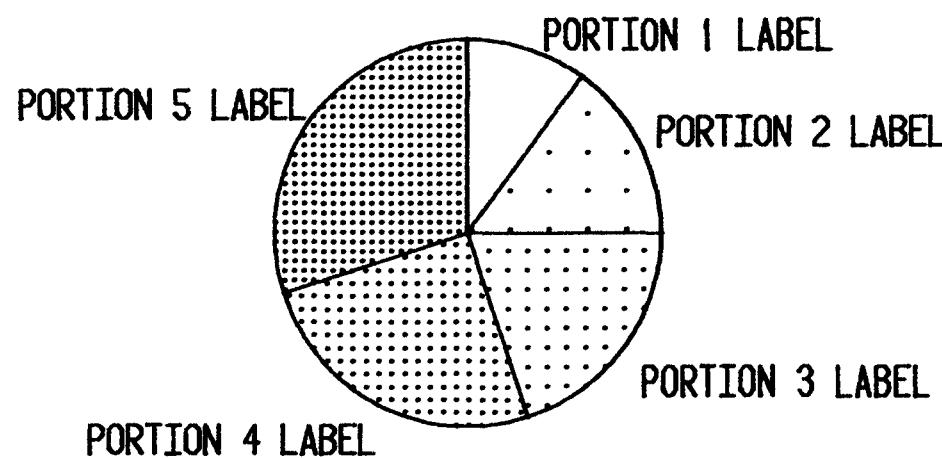


Figure 2.3. Example of an end product. The user-assigned labels here are actual terminologies used by the program to prompt inputs. The system sends the labels to corresponding locations according to these terms. All of the five available shades are displayed in these pies.

Labels

During the placement of pies, space may be set aside for labelling. The user-assigned labels of Figure 2.3 are the actual terminologies used by the program prompts. The labels are termed as 1) graph label, 2) pie label, and 3) portion label. Options are available to underscore the graph and pie labels as shown in Figures 2.2-2.4. Labelling can effectively be averted by entering a space followed by carriage return.

The graph label is placed directly above the rest of the graph, the pie labels are placed above their respective pies, and the portion labels are placed next to the corresponding portions of each pie. Clockwise beginning at the upper sector of the pie-circle, portion numbers are assigned consecutively from 1 to the total number of portions in each pie.

Character sizes may be selected for labels by specifying width and height, in that order. As an example for judging sizes, the respective character width and height for the graph label, pie labels, and portion labels are (2.5,4), (1.5,3), and (1.2,2.5) in Figure 2.2.

Data to be Represented

The input portion percentages must sum up to exactly 100 for each pie to avoid raising of the reentry flag by the program. In addition, negative percentages are not allowed.

Shades

The lines 2920 through 3120 as listed in Appendix A produce shades. The purpose of the shade option is to contrast the various portions of pies. The listed version of the program allows five different shades. The available shades are exhibited in Figure 2.3.

Table 2.2 Relationships among shade description, shade number, and relative distance between dots.

Shade Description	Shade Number	Relative Distance
Clear	1	-
Light Grey	2	2
Medium Grey	3	1
Dark Grey	4	2/3
Extra Dark Grey	5	1/2

Shades are produced by the placement of dots equally distanced from the nearest dots vertically and horizontally away. Dark shades are produced by close distances between dots. The relationships among shade description, shade number, and distance between dots are provided in Table 2.2. The relationship between shade number and distance between dots can also be described by the mathematical function: Distance = $2 \div (\text{Shade Number} - 1)$. The expression M(I)-1 in line 2920 of the listed program corresponds to the divisor of the above function, therefore, one can effectively change the distance between dots with alteration of that statement alone. Likewise, additional shades may also be included with related modifications.

The mechanism for dot placement involves a sequence of conditional statements. Whenever a portion of pie is to be shaded, the entire square circumscribing the pie-circle is scanned. The areas within the square but outside of the circle are eliminated from placement of dots by the execution of line 2990, which mathematically excludes those areas more than a radius distance away from the center of the circle. The lines 3000 to 3070 further exclude the area within the circle but outside of the portion of interest of the pie. Whenever a lattice point within the circumscribing square passes the series of sieving tests, a dot is placed on that spot.

In short, the mechanism is akin to first assessing the angular components of the polar coordinates of the points within the circle and then excluding those points that are not subtended by the immediate pie portion.

The technically oriented users may wish to experiment with different ways of forming shades by using various patterns of dots and other symbols. The shades on Figure 2.4 are formed by a staggered pattern of dots. In order to produce that effect, substitute the following lines for those of the listed program:

STAGGERED SHADES

PIE

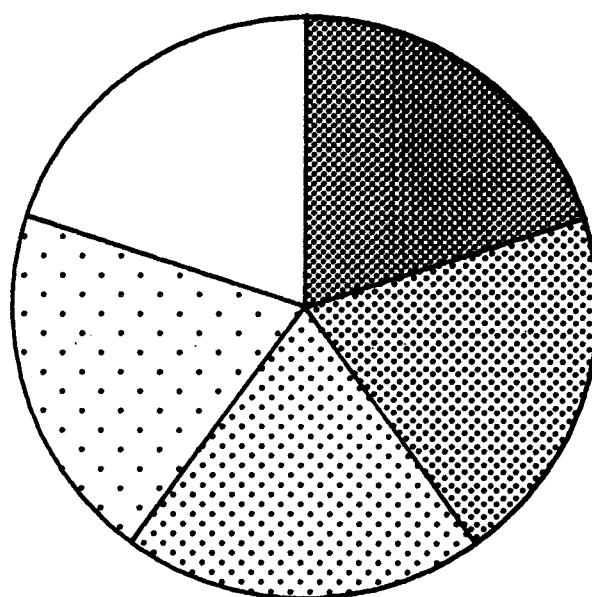


Figure 2.4. Example of a pie-chart shaded by staggered dots. This staggering pattern is accomplished with slight modifications of the program listed in Appendix A. Substitution lines are provided in the text.

2945 J3=-0.2
2947 J2=0
2970 R4=B(J)-(I1+J3)*R3
2980 R5=B(J)-(J1+J3)*R3
3122 IF J3=1 THEN 3130
3124 J2=1
3126 J3=0.3
3128 GO TO 2950

3. 3-D TRANSFORMATION

This program allows the representation of three-dimensional figures. The figures may be rotated, translated, and scaled interactively. When 3-d surfaces are plotted, there is an option to eliminate hidden lines.

The following is the user-definable-key (UDK) function list:

- 1) Counterclockwise rotation
- 2) Rotation along vertical axis
- 3) Rotation along horizontal axis
- 4) Translation up
- 5) Translation down
- 6) Translation left
- 7) Translation right
- 8) Scale larger
- 9) Scale smaller
- 10) Display of present transformed figure
- 11) Display of original figure and discard present transformation
- 12) Input data (tape or keyboard)
- 13) Input surface function
- 14) Set parameters and process surface function:
 1. Set x range and incrementation
 2. Set y range and incrementation
 3. Set hidden line option
- 15) Numeric display of original data
- 16) Numeric display of transformed data
- 17) Numeric display of minimum/maximum of transformed data
- 18) Save original data on tape

- 19) Save transformed data on tape
- 20) Set parameter:
 1. Degree of each rotation
 2. Center of rotation
 3. Translational units
 4. Scaling factors
 5. Display of center of rotation
 6. Display of rotated angles
 7. Output device
 8. Retention of transformations on screen

Scaling

Figure 3.1 demonstrates the usage of the scaling transformation. Each one of the successive images varies from its immediate neighbors by a factor of 1.4 in each of the three dimensions.

Rotation

Figure 3.2 demonstrates the usage of the counterclockwise rotation function. Each one of the successive images was rotated 45 degrees with the center of rotation indicated by a small circle at the center of the figure.

Figure 3.3 is a collection of images representing regular dodecahedrons (a solid figure with twelve faces, all of which being congruent regular pentagons) after having been rotated through various angles along vertical and horizontal axes.

Data Input

Figure 3.4 was constructed by specifying the vertices of the object

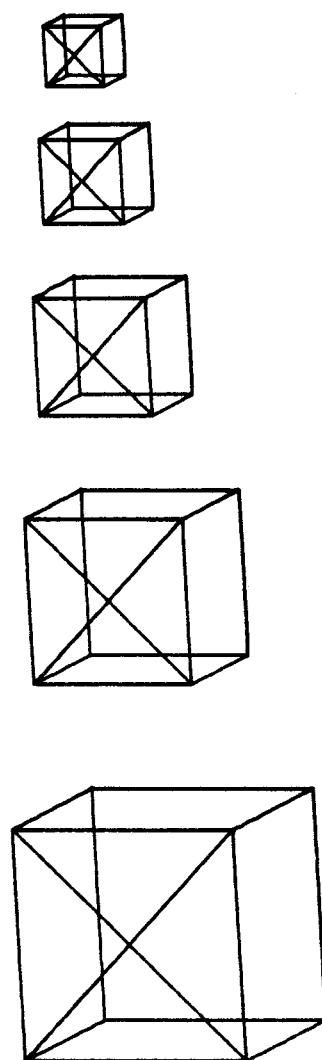


Figure 3.1. Scaling transformation. The successive images vary from their immediate neighbors by a scaling factor of 1.4 in each of the three dimensions.

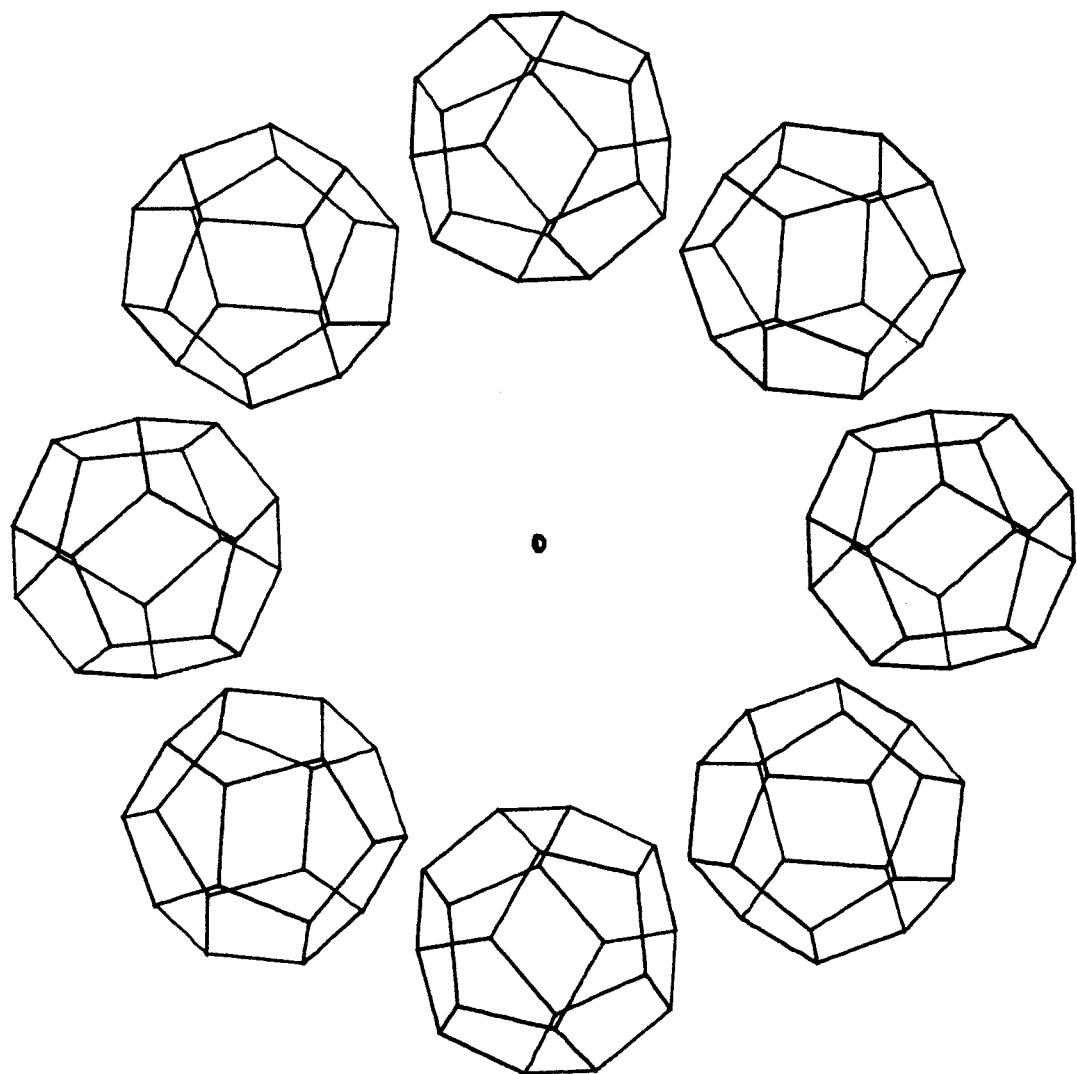


Figure 3.2. Counterclockwise rotation. The successive images are produced by rotating 45 degrees with the center of rotation indicated by a small circle at the center of the figure.

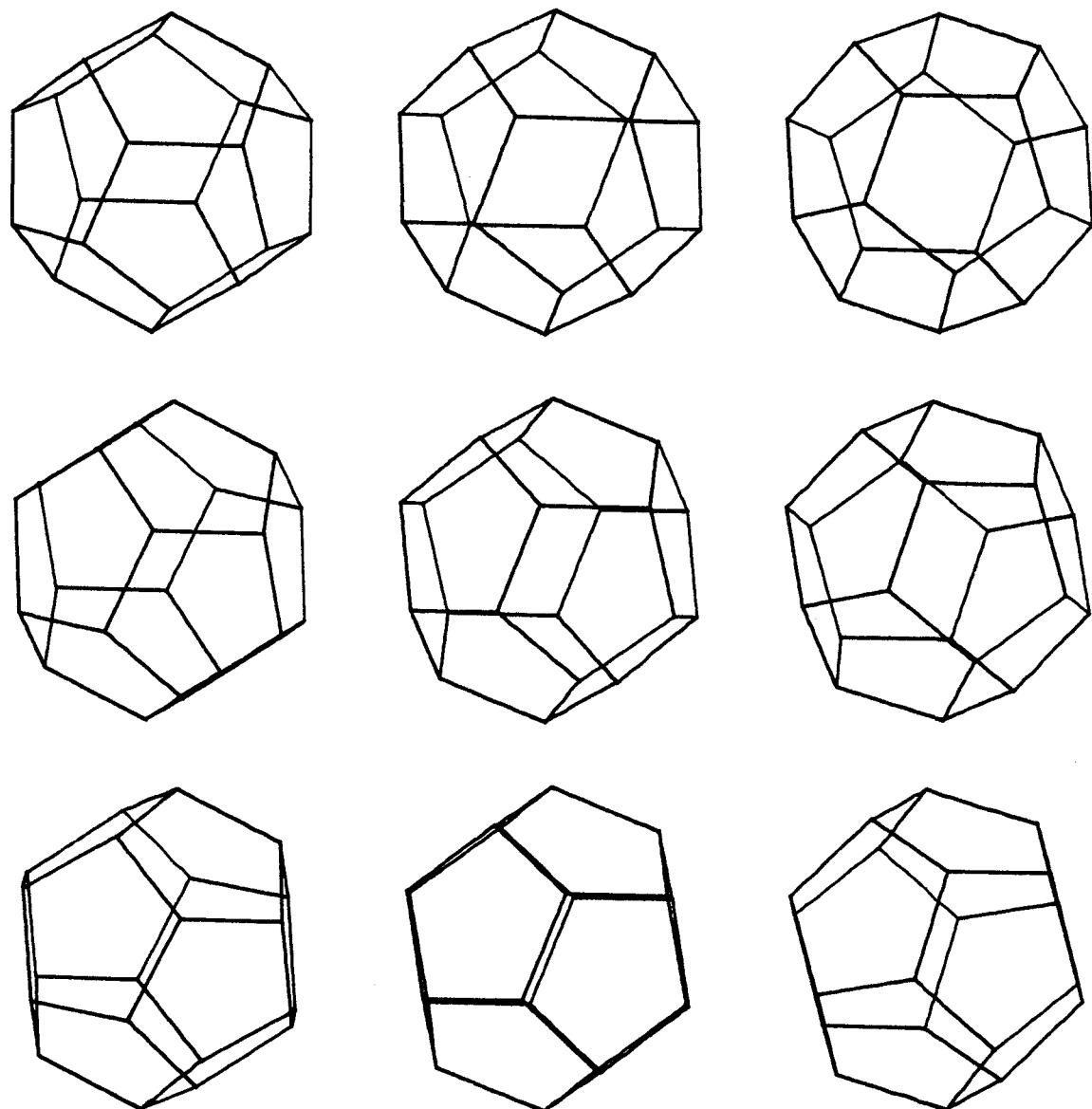


Figure 3.3. Collection of rotations. Each image represents a dodecahedron (a solid figure with twelve faces, all of which being congruent regular pentagons) viewed at various angles after having been rotated along the vertical and horizontal axes.

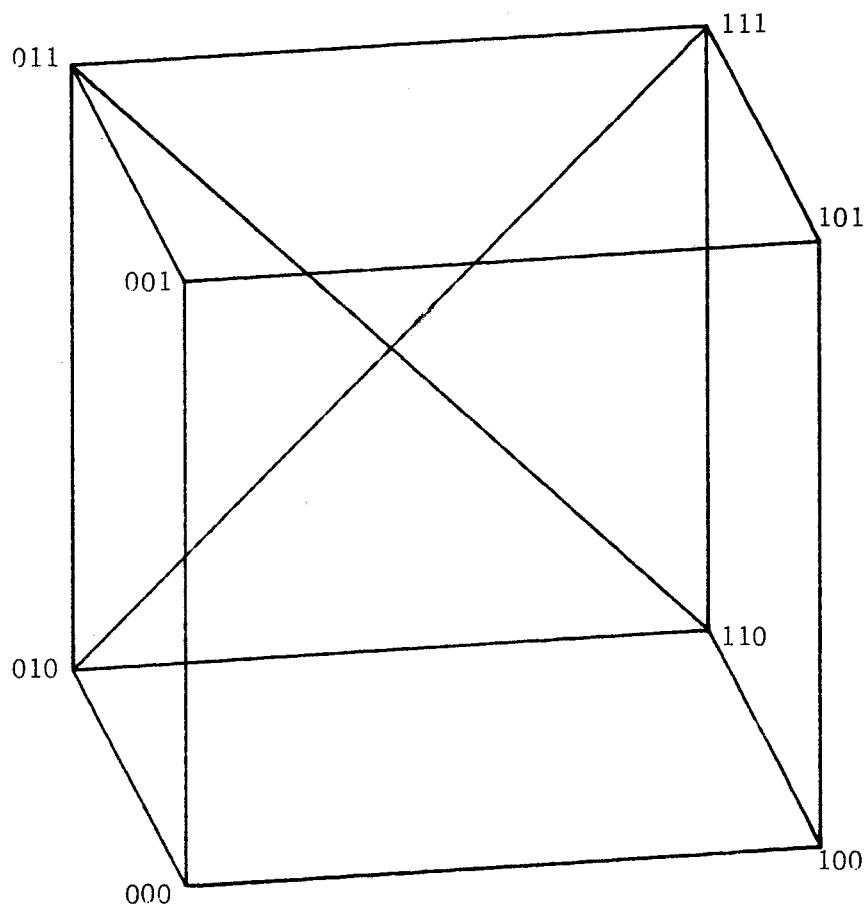


Figure 3.4. A construction example. The three digits at each vertex of the cubic figure represent the x, y, and z coordinates, respectively. The figure is constructed by entering the coordinates of the vertices and by stating whether the adjacent pairs of vertices should be connected by straight lines.

and whether each of the neighboring pairs of vertices is to be connected with a straight line. The following is the user-input used for the construction. Each entry represents the coordinates of a vertex of a cubic figure followed by an indication of whether the present point is to be connected to the previous one (Y is connect and N is disconnect).

```
(0,0,0) N
(1,0,0) Y
(1,0,1) Y
(0,0,1) Y
(0,0,0) Y
(0,1,0) Y
(1,1,0) Y
(1,0,0) Y
(0,0,1) N
(0,1,1) Y
(1,1,1) Y
(1,0,1) Y
(0,1,1) N
(0,1,0) Y
(1,1,1) Y
(1,1,0) Y
(0,1,1) Y
```

Figures 3.5 and 3.6 are two perspectives of a surface generated by the function $Z = | 10000 \cdot \cos X \cdot \cos Y / XY |$. The surface peaks in the neighborhood of $(x,y) = (0,0)$. As one can see upon examination of the function, the function value approaches infinity as (x,y) approaches $(0,0)$. The ranges in degrees at which the surface is being viewed are from -150 to 150, and from -90 to 210 for the x and y coordinates, respectively. Increment in both the x and y directions is 20 degrees. (The incrementation parameter specifies the distance between lattice points.) Lattice points are identified by the x and y coordinates where function values are evaluated. In general, a smaller incrementation gives a smoother and finer resolved surface.

Figure 3.7 is generated by the inclusion of the following lines:

```
3150 X5=SQR(X5)
```

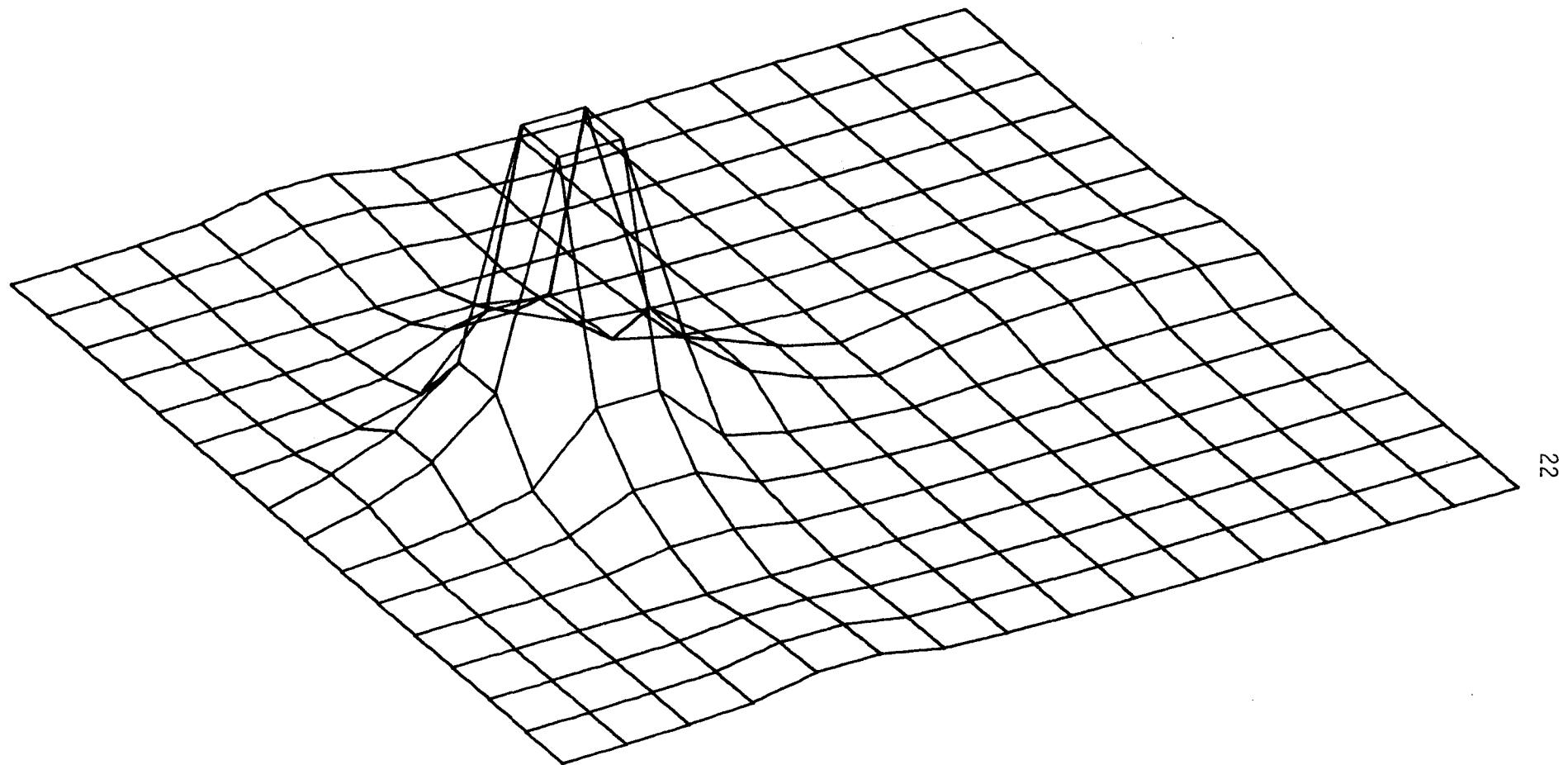


Figure 3.5. A perspective of a surface. This figure is produced by entering a mathematical function of a three-dimensional surface, the ranges in the x and y directions, and the grid size.

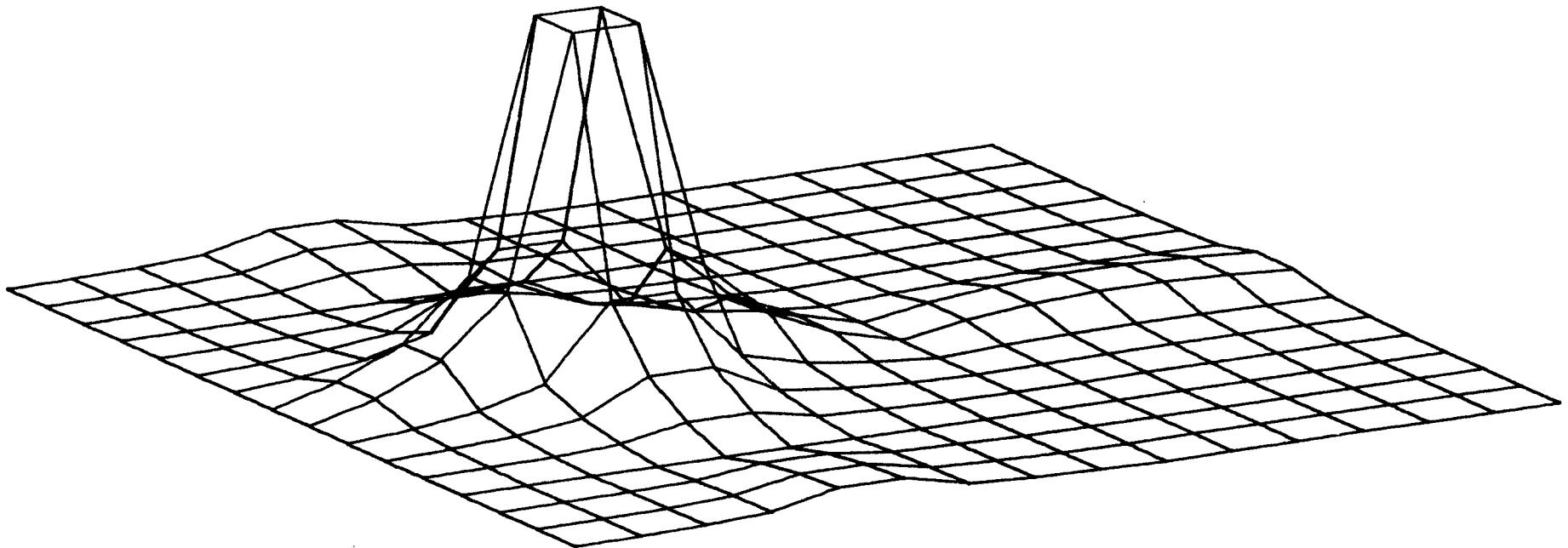


Figure 3.6. A perspective of a surface. This figure is the same as that of Figure 3.5. It has been transformed through the rotation functions, therefore, the surface is being viewed at a different perspective.

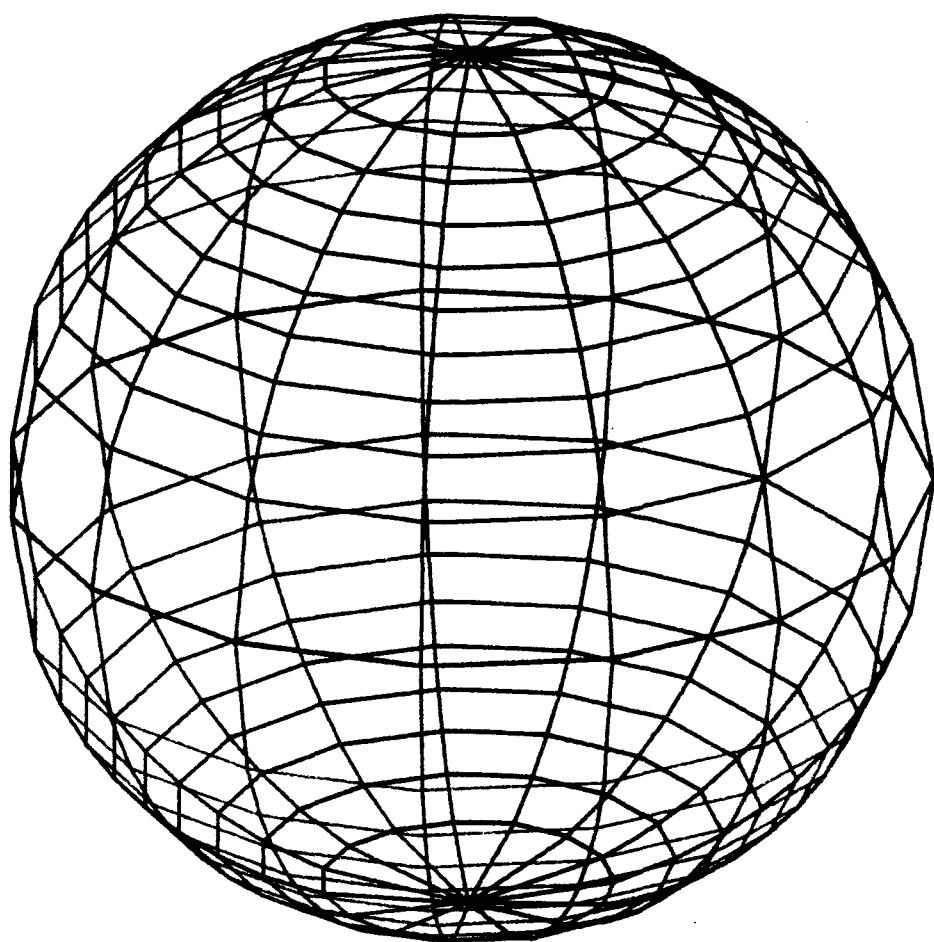


Figure 3.7. A plot generated by specifying the function for half of a sphere. The total figure is formed by two half-spheres. The difference between the two halves of the sphere is a rotation along a horizontal axis by 180 degrees.

```

3151 Z5=SQR(1-X5↑2)
3152 Y6=Y5
3153 Y5=X5*SIN(Y5)
3154 X5=X5*COS(Y6)

```

The input x range is (0,1) with an incrementation of 0.1, and the y range is (0,360) with an incrementation of 24. Only half of the sphere can be generated with the above specifications. The remaining half was produced by a rotation of 180 degrees along a horizontal axis.

Hidden Lines Removal

Figure 3.8 depicts two surfaces with identical function and viewpoint. The lower of the two surfaces has its hidden lines removed. The hidden line algorithm being used here is quite involved and it consumes an enormous amount of processing time. For example, with its 110 lattice points, it took approximately three hours to complete the calculations necessary to draw the lower surface of Figure 3.8. Processing time is roughly a square function of the number of lattice points. That is, doubling the number of lattice points would approximately quadruple the processing time.

The algorithm is essentially divided into two parts. In order to ascertain whether a line segment is to be plotted from one lattice point to the adjacent lattice point, the first part of the algorithm is required to determine all the points of the line segment where it is intersected by other line segments as projected to a two-dimensional plane. These points partition the line segment into sub-segments. These sub-segments are sent to the second part of the algorithm which determines whether each of the sub-segments is hidden by the faces of the surface where a face is defined to be a quadrilateral formed by four adjacent lattice points. In

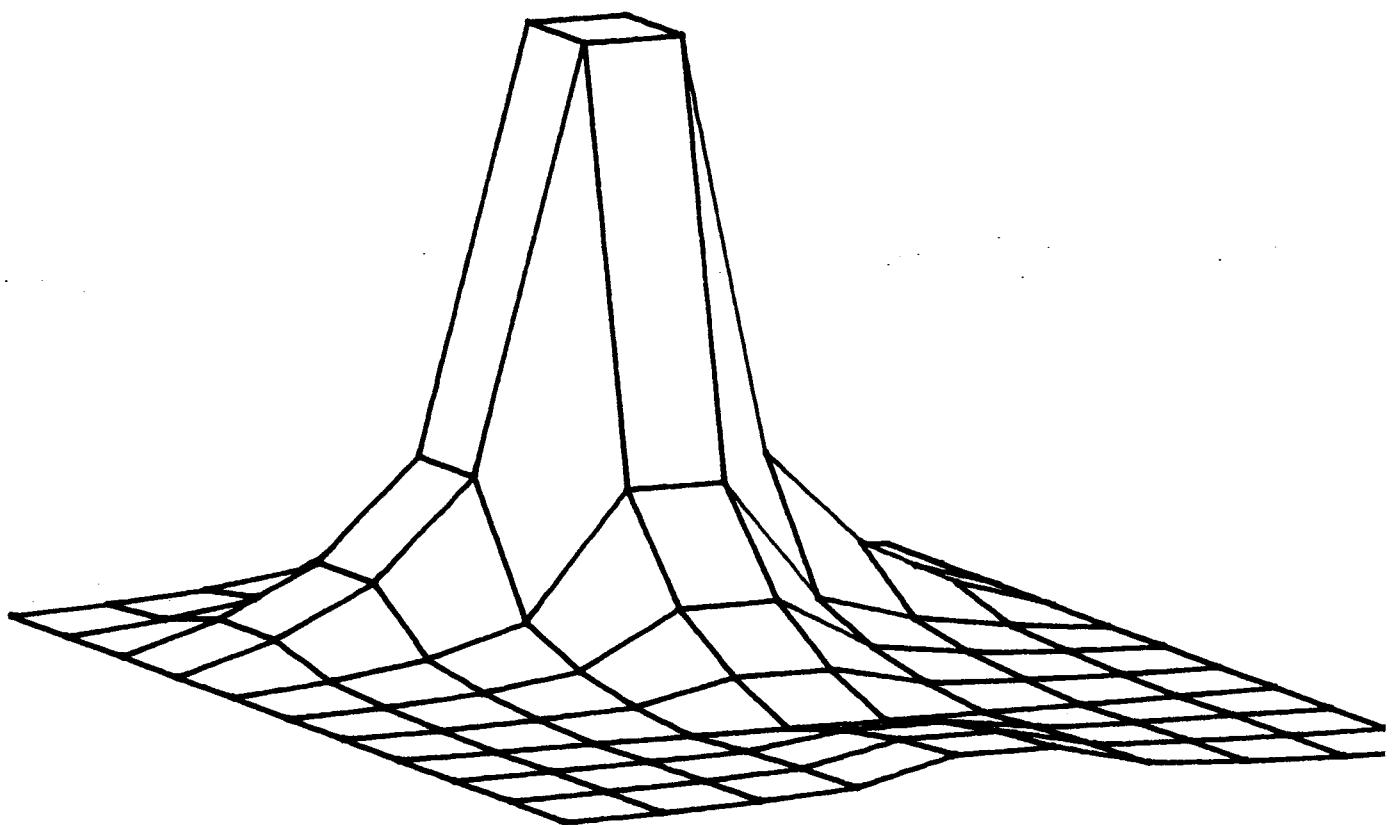
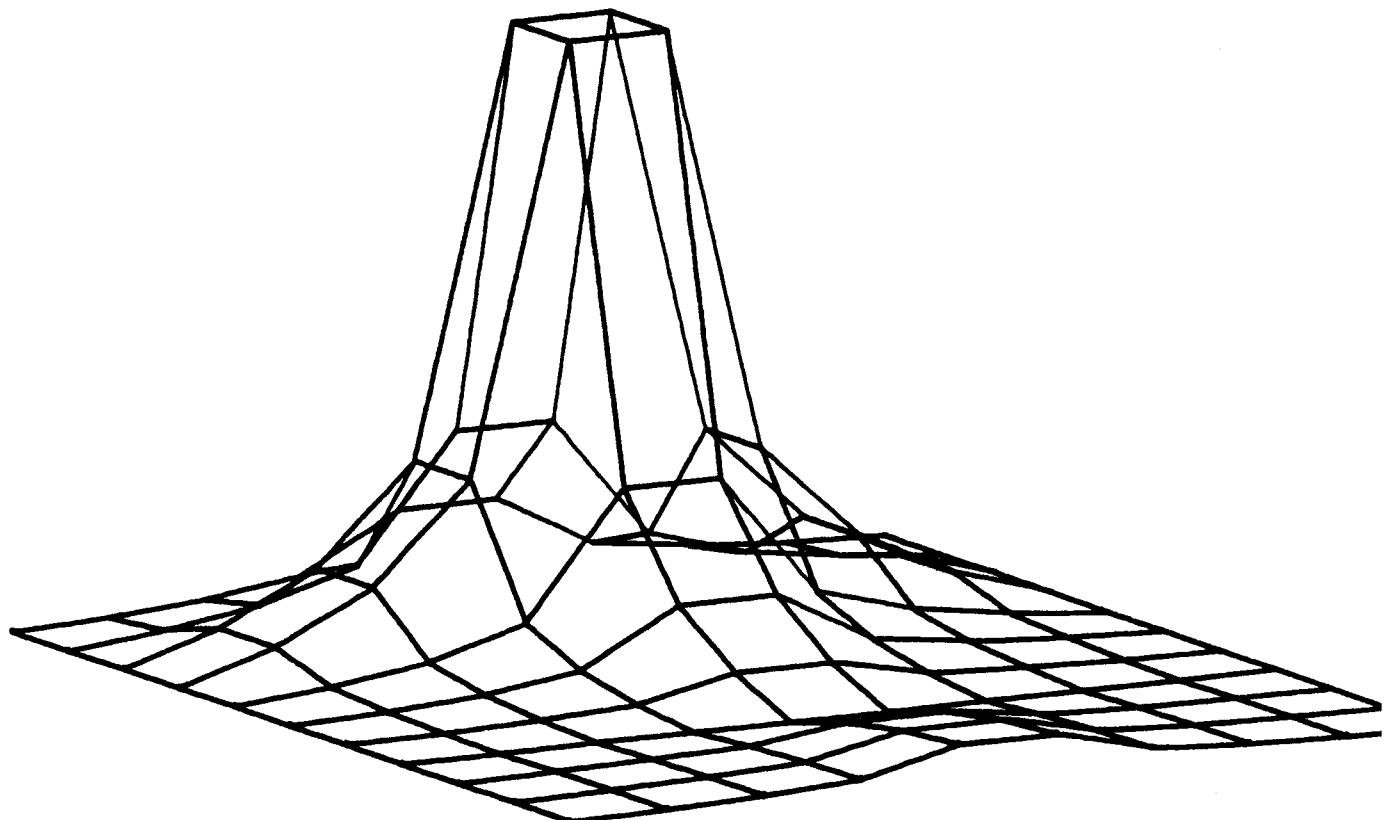


Figure 3.8. Hidden line option. These two surfaces are identical except that the lower of the two has its hidden lines removed.

order to make this determination, it can be demonstrated that all the program needs to consider is whether any single point within the sub-segment is hidden. A point is hidden or covered when it is enclosed by a face which is closer to the viewer than that point. Obviously, if a point is hidden by a face, then the entire sub-segment to which the point belongs is also hidden by the same face. Therefore, only one point of each sub-segment needs to be considered. I have chosen the midpoint of each sub-segment to be that point.

Each of the quadrilateral faces is divided, not partitioned, into four triangles in order to ascertain its surface feature. For example, for a given quadrilateral ABCD, one obtains triangles ABC, BCD, CDA, and DAB. Each of the triangles identifies a planar surface that intersects all three of the triangle's vertices. The surface feature of ABCD is described, roughly, by these four intersecting planes.

Let us define that point P is enclosed by a three-dimensional geometrical figure if and only if point P is within the boundaries of the geometrical figure as projected to the line of sight of the viewer. Consider the point P of a sub-segment that is enclosed by both the triangles ABC and DAB in Figure 3.9. The planes formed by the triangles are in general not identical. Point P is considered to be hidden by this algorithm if either one of the two planes covers P. And, as mentioned earlier, if P is covered, then the entire corresponding sub-segment is also covered.

The main body of the second part of the algorithm consists of the lines 6450 through 6900 listed in Appendix B. This relatively short subroutine encases a rather extensive decision tree. Examination of the various branches of the tree can be helpful in understanding the algorithm and its construction. The algorithm tests each face of the surface to

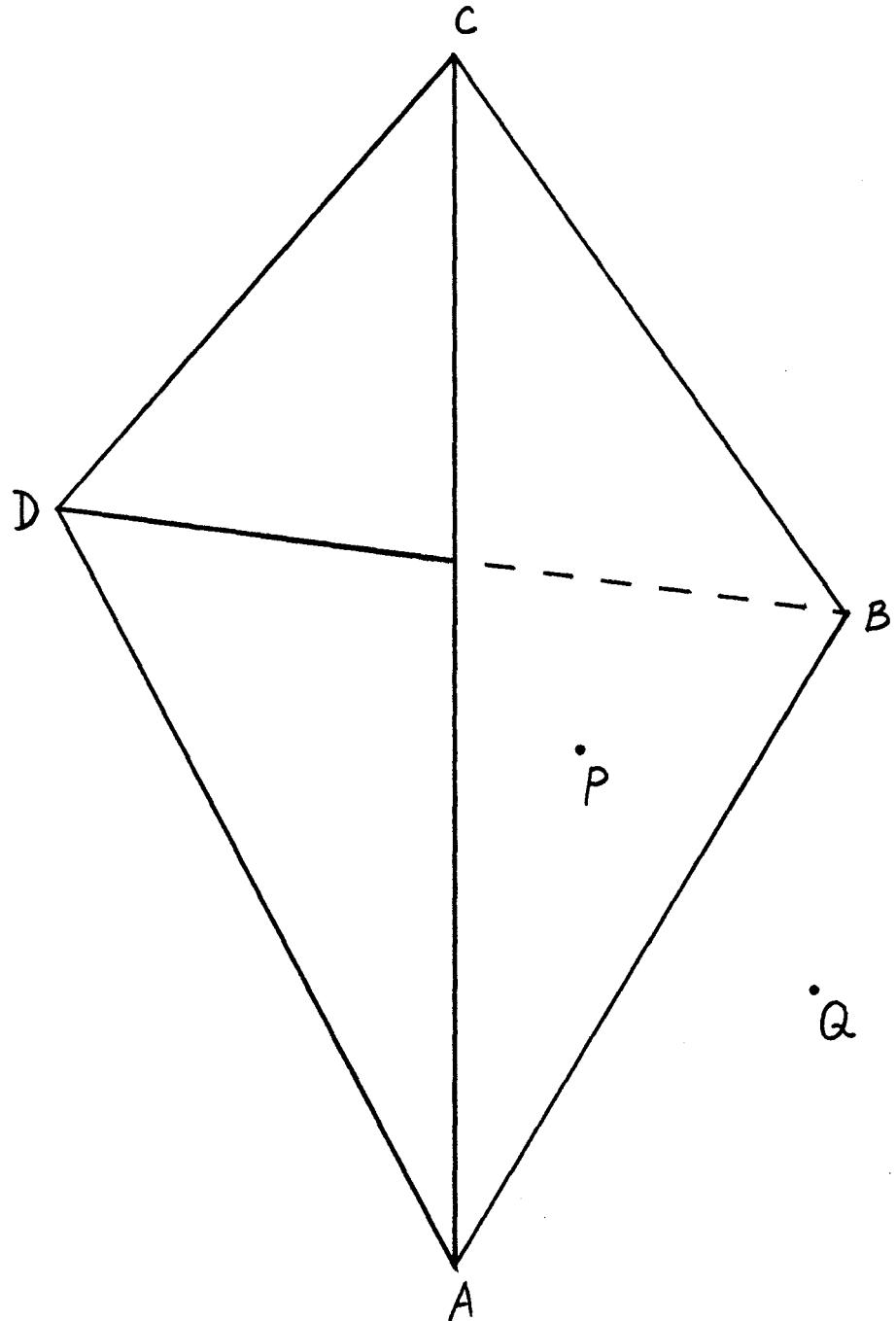


Figure 3.9. A model for explaining the hidden line algorithm. See the "Hidden Lines Removal" subsection of text for complete explanation.

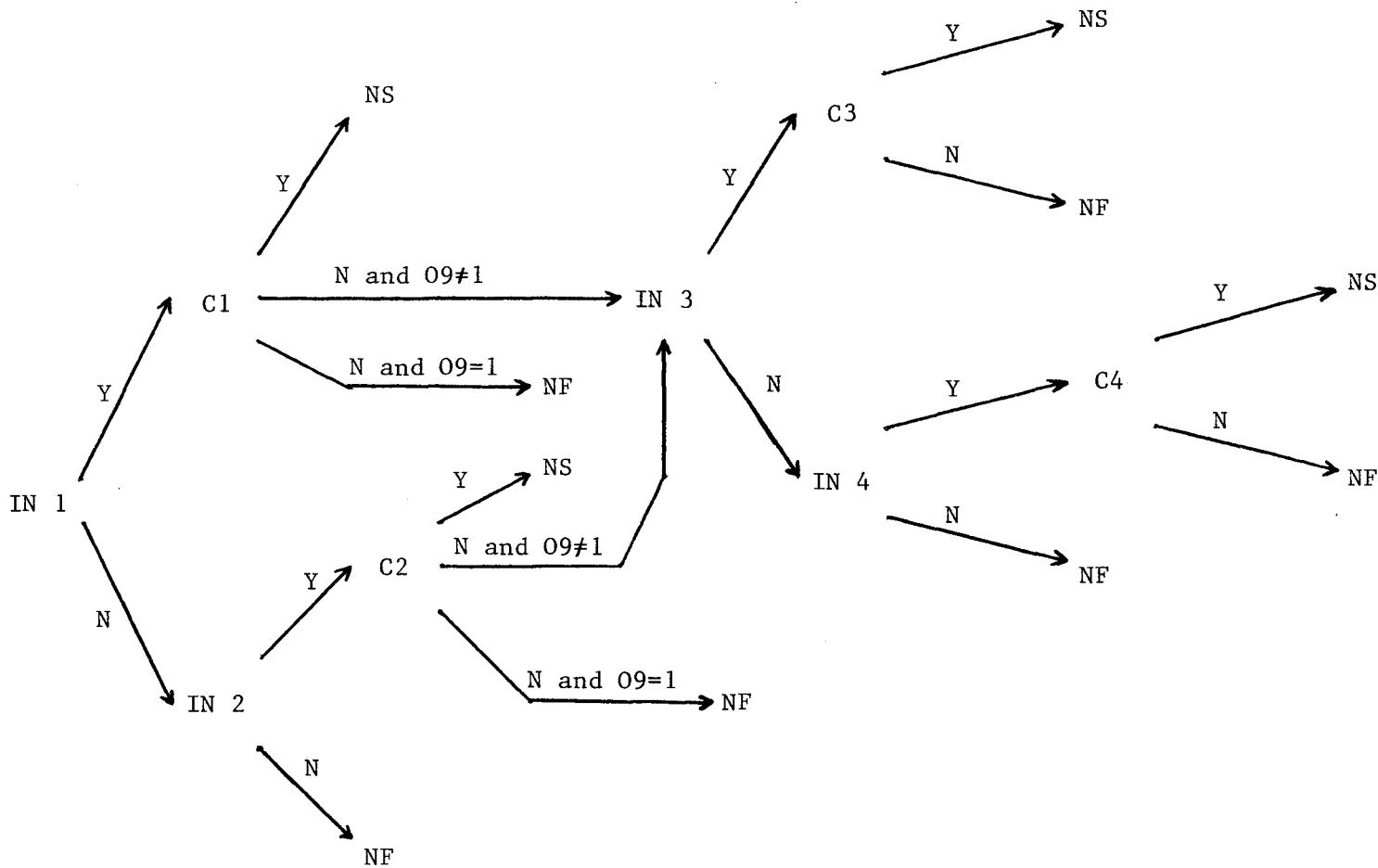


Figure 3.10. A decision tree embedded within the hidden line algorithm. The four triangles that divide a face are denoted by 1, 2, 3, and 4. The abbreviations 'IN X' and 'CX' at the branching points, where $X = 1, 2, 3, 4$, are determinations of whether the midpoint of the sub-segment is enclosed in triangle X , and whether the midpoint is covered by triangle X , respectively. When the convex hull of the vertices of a quadrilateral face is not congruent to the quadrilateral face, only two of the four triangles need to be considered for evaluation. This special case is identified by the variable name 09 having the value of 1. As usual, 'Y' stands for yes, and 'N' for no. Next segment and next face are abbreviated as 'NS' and 'NF', respectively.

ascertain whether a sub-segment is to be plotted. The decision tree makes several determinations and decisions. First, if a face does not enclose or encloses but does not cover the sub-segment, then the subroutine branches to the next face. Second, if after all the faces have passed through the paths of the decision tree and each of them either does not enclose or encloses but does not cover the sub-segment in question, then that sub-segment is drawn. Third, if a face is found to enclose and cover the sub-segment, then the sub-segment is not drawn, and the subroutine goes on to consider the next sub-segment. The decision tree algorithm is graphically displayed in Figure 3.10.

The mathematics involved to implement the algorithm is elementary. It includes determining the equation of the line that intersects two specified points, the point at which two specified lines intersect, the equation of the plane that intersects three specified points, and the area of the triangle with known vertices. The latter is used for identifying whether a point is enclosed within a triangular area specified by three lattice points. For example, in Figure 3.9, we can conclude that P is enclosed within triangle ABC by the equality of the total sum of triangular areas PAB, PBC, and PCA with the triangular area ABC. The point Q is not enclosed because the combined areas of QAB, QBC, and QCA are greater than the area ABC.

4. BARGRAPH

This program produces bargraphs such as those in Figures 4.1-4.3. Multivariate data consisting of one continuous variable in addition to up to four discrete variables may be represented. The continuous variable can be represented by its means and standard deviation such as in Figures 4.1 and 4.2. It can also be represented by a single value such as in Figure 4.3. The discrete variables are distinguished by the denotations: treatment, group, block, and story. These denotations are shown in Figure 4.2 by their corresponding labels. Treatments are distinguished by various bar enhancements. The discrete variables may be assigned according to the above terminology depending on the user-desired output format, arrangement of data for meaningful representation, and priority of comparisons among discrete variables.

Input Variables

Table 4.1 is a list of input variables and their corresponding names. The variable values may be listed and altered by the user when the program is not in busy mode. The busy mode or the interactive session can be exited by typing the break key twice or after a graph is plotted. For an example of a modification, let us suppose the user wishes to alter the width of the graph label characters without reentry of the entire data set. The user must first exit the busy mode, then enters Z2, the variable name for width of graph label characters. The system then responds with the value that resides in Z2, e.g., 2.5. In order to increase the width to 3, the user types Z2=3, and the change is completed. Contents of string variables, such as P\$, B\$, G\$, X\$, and Z\$, should be placed within quotation marks,

GRAPH LABEL

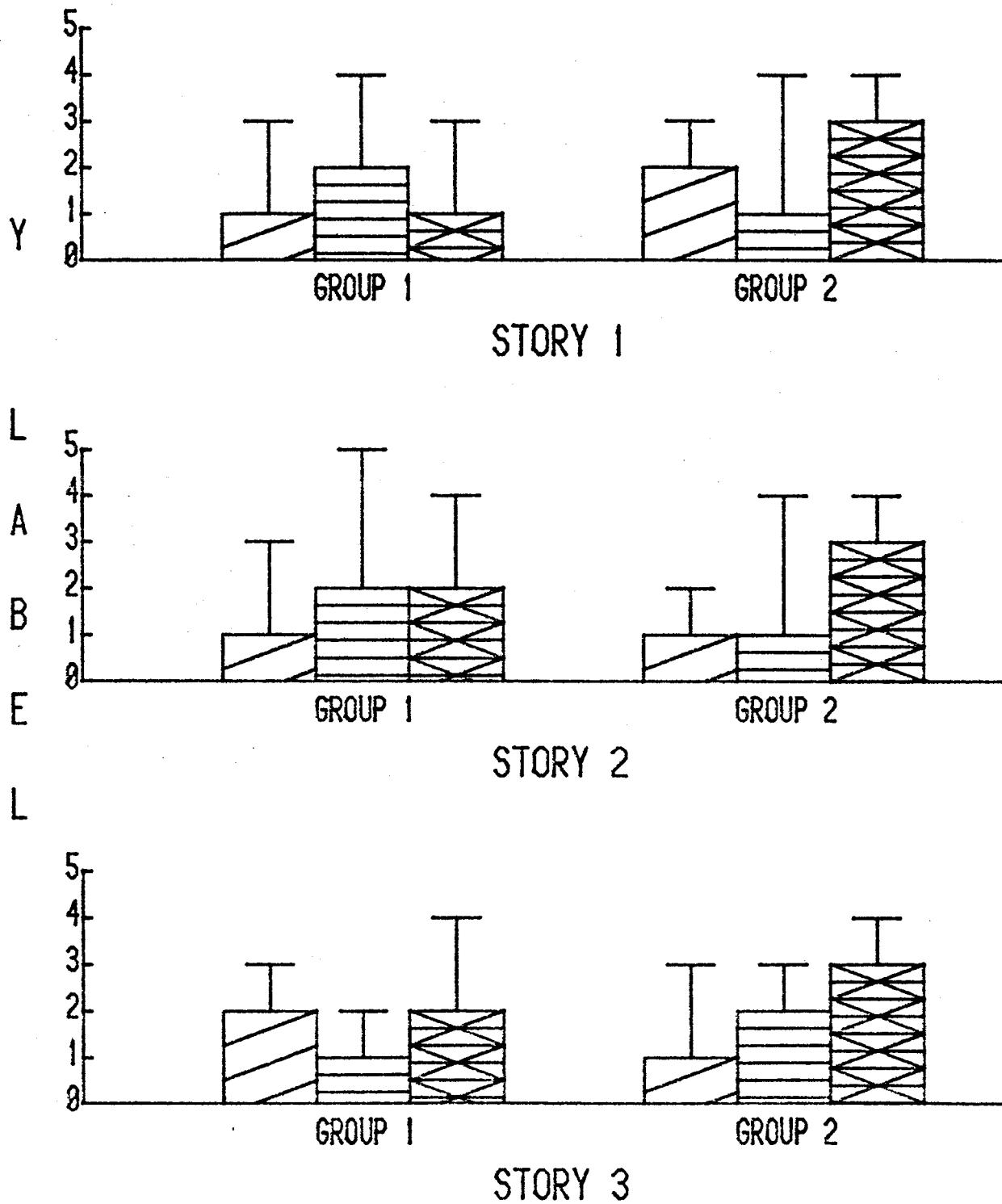


Figure 4.1. A bargraph example. This bargraph has three stories, one block each story (the plotting of block labels is averted by a blank entry), two groups in each block, and three treatments in each group.

GRAPH LABEL

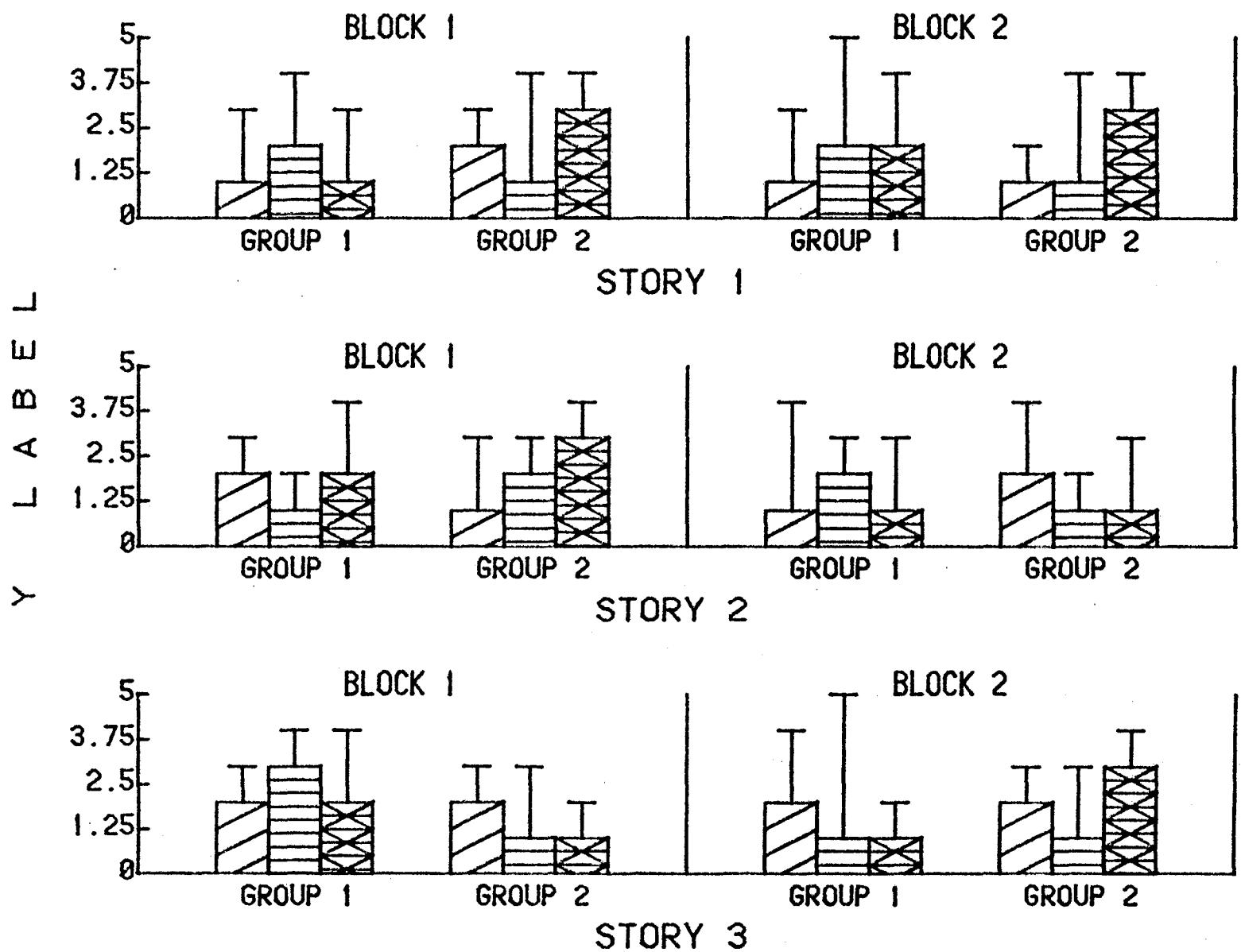
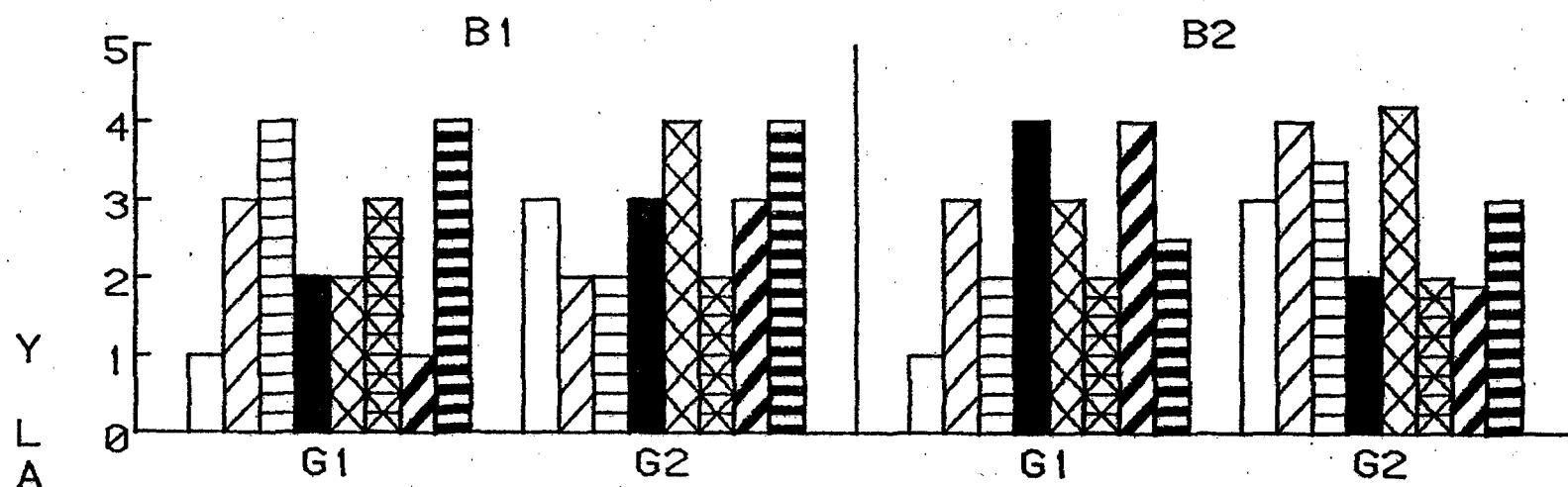


Figure 4.2. A bargraph example. This bargraph has three stories, 2 blocks in each story, 2 groups in each block, and three treatments in each group. The y axis label is rotated 90 degrees.

DEMO GRAPH



34

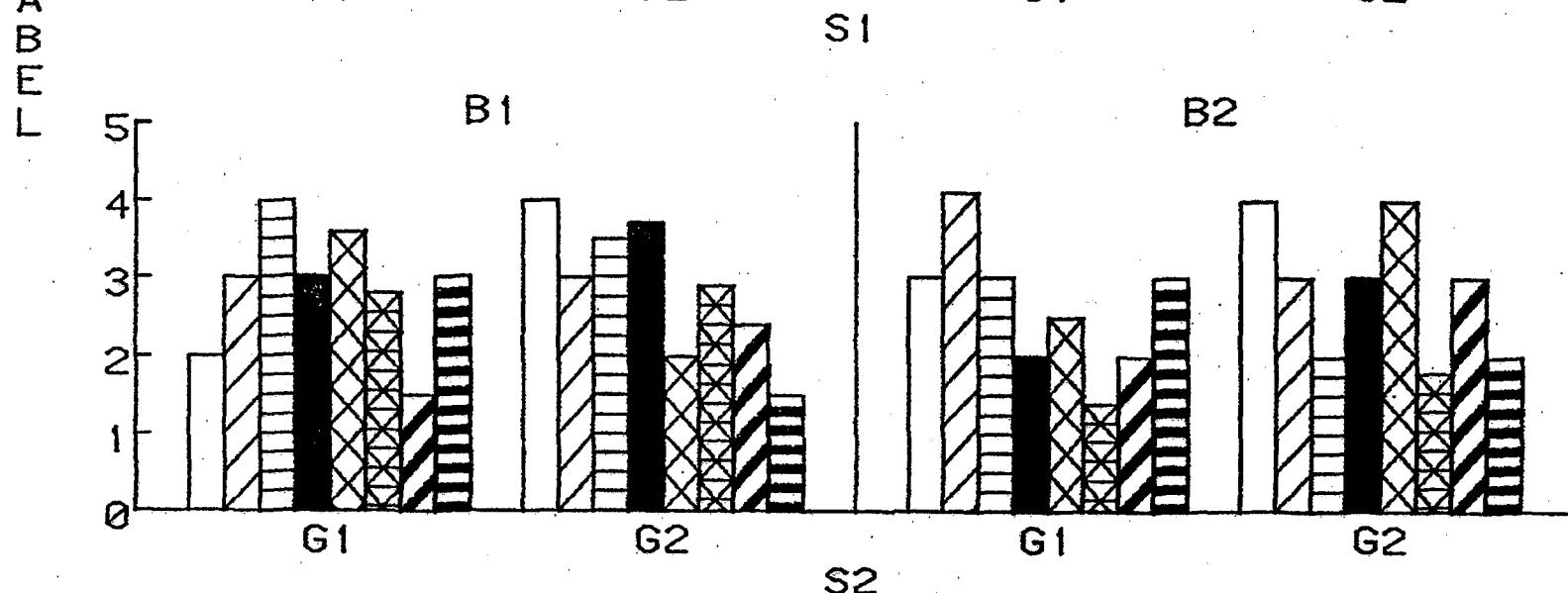


Figure 4.3. A bargraph example. This bargraph has two stories, two blocks in each story, two groups in each block, and eight treatments in each group. All of the eight enhancements available for treatment identification are presented.

Table 4.1 List of input variables: programmer-assigned variable names and their corresponding descriptions.

Programmer-assigned Name	Variable Description
V1	Minimum of continuous variable
V2	Maximum of continuous variable
V6	Tic mark increment on y axis
V7	Width of character size for y axis numerals
V8	Height of character size for y axis numerals
S6	Total number of stories
P\$	Story labels
Y2	Width of character size for story labels
Y3	Height of character size for story labels
B1	Total number of blocks at each story
B\$	Block labels
B2	Width of character size for block labels
B3	Height of character size for block labels
G1	Total number of groups in each block
G\$	Group labels
G2	Width of character size for group labels
G3	Height of character size for group labels
T1	Total number of treatments in each group
K(I)	Corresponding number of the enhancement for treatment I
X\$	y axis label
X2	Width of character size for y axis label
X3	Height of character size for y axis label
Z\$	Graph label
Z2	Width of character size for graph label
Z3	Height of character size for graph label
D	Values of continuous variable corresponding to treatments, groups, blocks, and stories
L	Standard deviations associated with D above
O	Output device code

for example, P\$="STORY 1". Then, to have the graph plotted again, but now with the changes, push the user-definable-key 1 at the upper-left corner of the keyboard.

Labels

The user has the option to label each variable as well as the graph. The character sizes for the labels as well as for the numerals on the y axis may also be specified. (Note: Character size is fixed on screen outputs; 1.79 for width and 2.82 for height.) For plotter outputs, the characters of the y axis label (the continuous variable) may be rotated 90 degrees such as that in Figure 4.2.

Enhancements

Seven enhancements, in addition to blank (no enhancement), are available to distinguish bars of different treatments. The enhancement names are listed in Table 4.2.

All the enhancements are exhibited in Figure 4.3 respective to the order listed in Table 4.2.

What the enhancements represent in the user's graph can be described by using the "Legend Maker" program presented in Section 6. The "Legend Maker" program can be used independently after the bargraph is plotted. The legend can be placed anywhere on the graph by specifying the lower-left and upper-right margins of the plotter.

Table 4.2. Enhancements and their corresponding numbers.

Number	Description
1	Blank (no enhancement)
2	Cross-hatching
3	Horizontal bars
4	Solid
5	Crosses
6	Stars
7	Striated cross-hatching
8	Striated horizontal bars

5. FRACTIONAL BARGRAPH

The construction of this program is similar to that of the "Bargraph" program described in the previous section. Instead of having treatment as the continuous variable, "Fractional Bargraph" has segment (or fraction). All other variables -- group, block, and story -- serve the same functions as those in "Bargraph."

This program is suited for representation of data when the values of the quantitative variable sum up to a fixed number. For example, the concentrations of protein, moisture, fat, and ash from proximate analysis of fish samples, which sum up to 100%, can be suitably represented by a fractional bargraph. Each fish sample may be represented by a stack of four segments, with each segment representing the percentage of protein, moisture, fat, or ash. The effect is a sort of hybrid between the bargraph and the pie-chart. Figure 5.1 is an example of the "Fractional Bargraph" outputs.

Input Variables

The input variable list is similar to that of "Bargraph." The exceptions are that the minimum of the continuous variable, named V1 in "Bargraph," is fixed in this program at 0; V2 now stands for the summation of segments in each group ($V2=100$ in the above fish sample example); and, T1 represents the number of segments in each group, whereas in "Bargraph," it is the number of treatments in each group.

The content of each variable can be listed and altered when the system is not in the busy mode. The procedure is detailed in the "Bargraph" section.

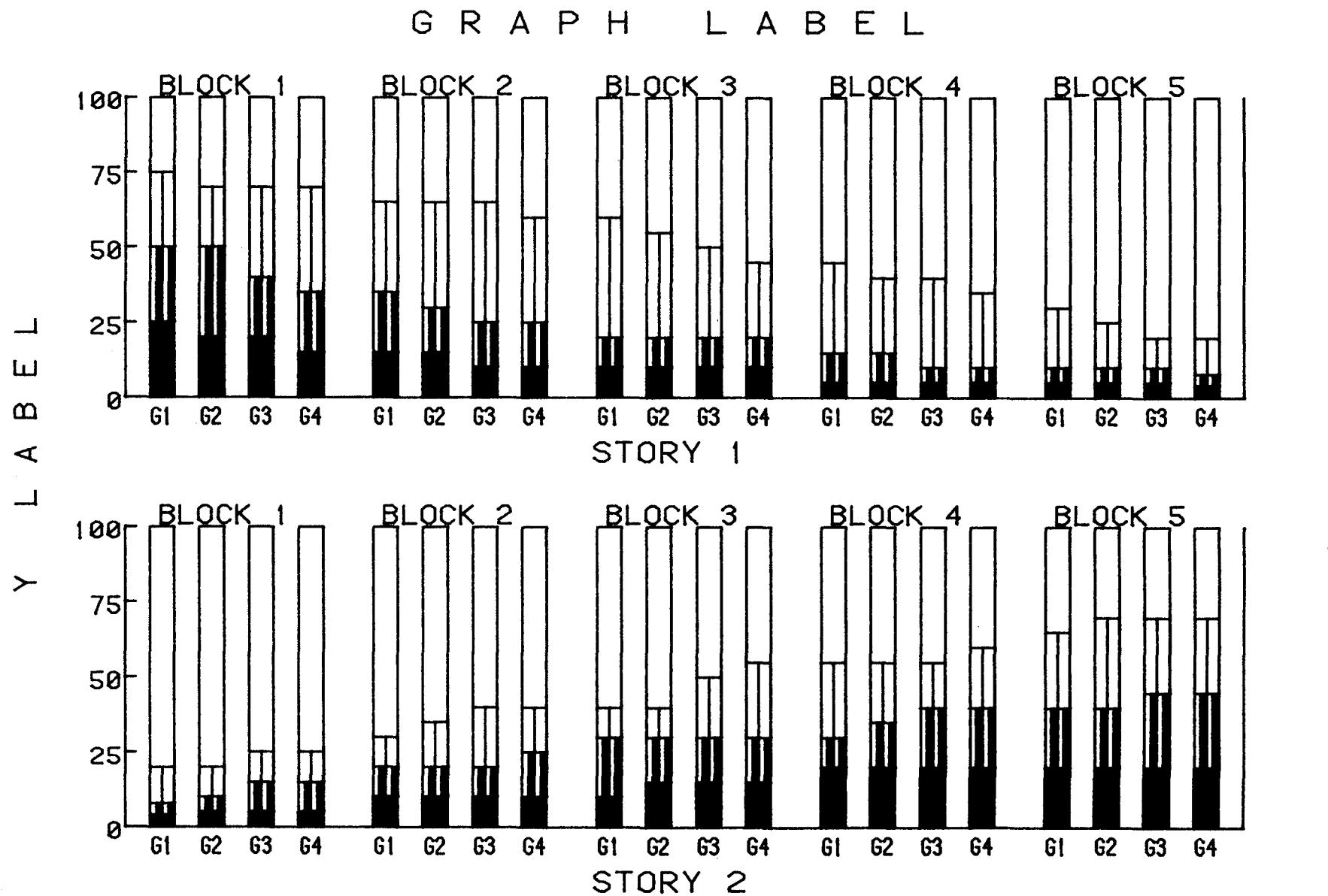


Figure 5.1. A fractional bargraph example. This graph has two stories, five blocks in each story, four groups in each block, and four segments in each group. All of the four available enhancements for identifying segments are exhibited. The y axis label is rotated 90 degrees by an affirmative response to the corresponding option prompted during the interactive session.

Enhancements

Three enhancements, in addition to blank (no enhancement), are available to distinguish among segments. All four enhancements are exhibited in Figure 5.1. The enhancement names are listed in Table 5.1.

The "Legend Maker" presented in the next section can be used to describe what the user-assigned enhancements represent as it can similarly be used to describe the enhancements of bargraphs of the previous sections.

Table 5.1. Enhancements and their corresponding numbers.

Number	Description
1	Blank (no enhancement)
2	Stripes
3	Vertical lines
4	Solid

6. LEGEND MAKER

This program is written as a supplement to the "Bargraph" and the "Fractional Bargraph" routines described in the two previous sections. "Legend Maker" plots legends as specified by interactive user-inputs.

Enhancements and Captions

A legend can be divided into two parts: enhancements and captions. The user may specify up to ten different enhancements: 1) blank, 2) cross-hatching, 3) horizontal bars, 4) solid, 5) crosses, 6) stars, 7) striated cross-hatching, 8) striated horizontal bars, 9) vertical lines, and 10) stripes. All of these enhancements are illustrated in Figure 6.1. Corresponding to each enhancement is a caption. Captions are texts that describe what the enhancements represent. Spaces can be inserted between an enhancement and its corresponding caption by entering spaces before the text during the input of the caption.

Size and Location

The sizes and locations of legends on the plotter outputs are largely determined by the margin settings of the plotter. The legend would fill up the set vertical margins, while the horizontal length of the legend is dependent on 1) the set horizontal margins, 2) the number of characters in each line, and 3) the user-input character width. The total width of a line can be evaluated as a fraction of the horizontal margin which encompasses 150 units. For example, if there are twenty characters in a line (including all the spaces and the enhancement symbol), and if the user entered 5 units for the character width during the interactive session,

- CAPTION FOR BLANK
- ▨ CAPTION FOR CROSS-HATCHING
- ▨ CAPTION FOR HORIZONTAL BARS
- CAPTION FOR SOLID
- ▨ CAPTION FOR CROSSES
- ▨ CAPTION FOR STARS
- ▨ CAPTION FOR STRIATED CROSS-HATCHING
- ▨ CAPTION FOR STRIATED HORIZONTAL BARS
- ▨ CAPTION FOR VERTICAL LINES
- ▨ CAPTION FOR STRIPES

Figure 6.1. A legend example. All ten of the available enhancements are illustrated. The names of the enhancements are given in the corresponding user-input captions of this example.

then the width of the line can be computed as 20×5 units = 100 units long, or, $100 \div 150 = 2/3$ of the horizontal margins.

The legend can be relocated on the plotter with the dimensions intact by resetting the lower-left plotter margin only, without resetting the upper-right margin.

Input Variables

Table 6.1 is a list of the input variables and their corresponding programmer-assigned names.

Table 6.1. List of input variables: programmer-assigned variable names and their corresponding descriptions.

Programmer-assigned Name	Variable Description
N	Total number of enhancements being used
E(I)	Enhancement number of the I(th) enhancement
L(I)	Number of characters in the I(th) caption
C\$	Caption text
W	Width of characters
O	Output device number

7. X-Y GRAPH

The original version of this program is a part of the manufacturer's Plot-50 System Software. The documentation and instruction on the usage of the original version can be found in the "4050 Series Operator's Manual." The present version of the program is modified to include additional capabilities.

Original Capabilities

The following is a summary list of the capabilities of the original version:

- 1) Plotting of points specified by two-dimensional cartesian coordinates, i.e., x-y plot
- 2) Option to connect the two-dimensional points with straight lines
-- line plot mode
- 3) Option to set x and y screen positions
- 4) Option to set x and y ranges
- 5) Five symbols for plotting of the two-dimensional points: point, triangle, plus sign, square, and diamond
- 6) Ability to edit data: insert, delete, and change
- 7) Data entry from either the keyboard or magnetic tape
- 8) Data storage onto magnetic tape.

Added Capabilities

The following is a list of additional capabilities included in the present version of the program:

- 1) Capability to output graphs on plotter

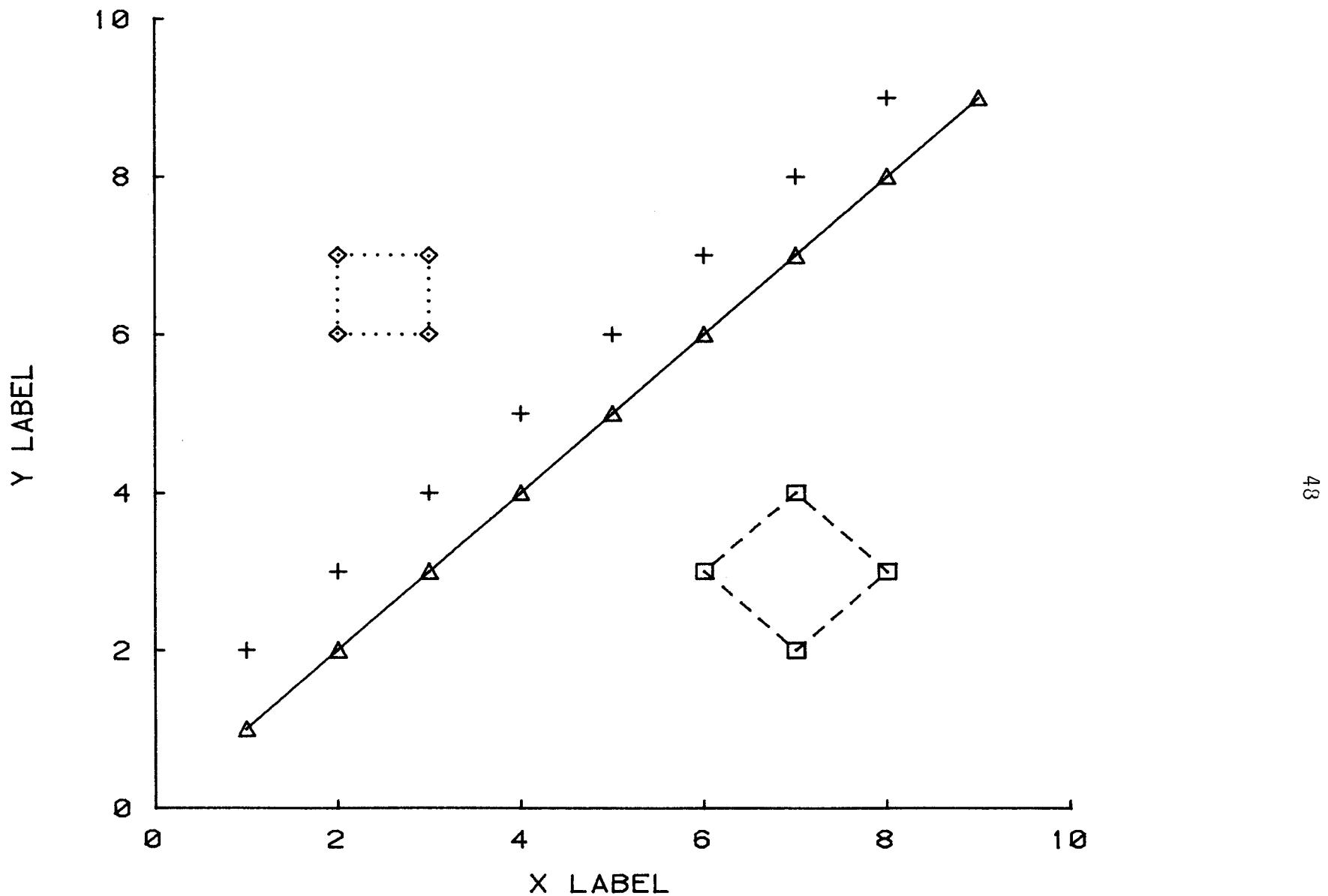
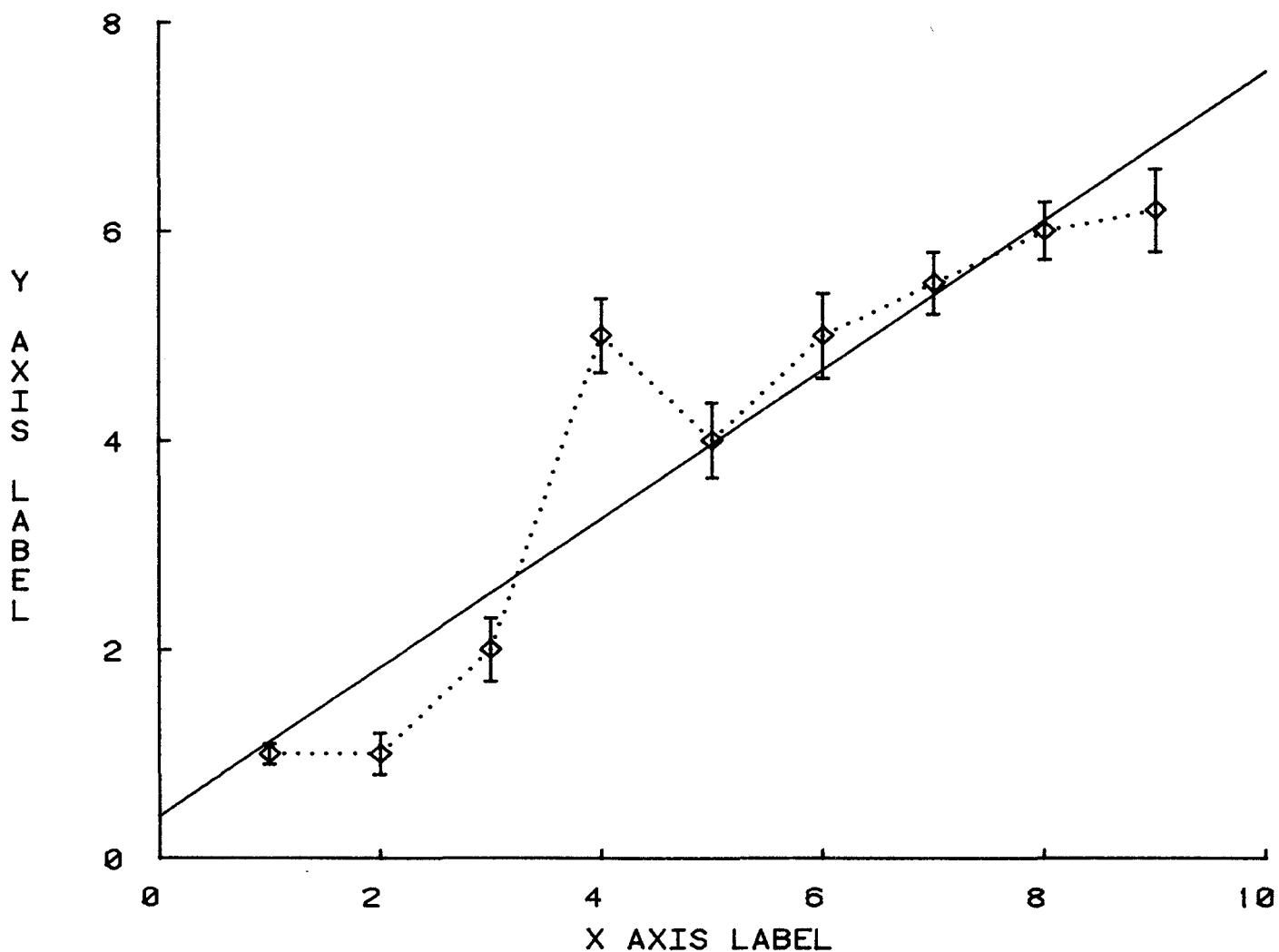


Figure 7.1. An example of x-y graphs. All four of the available plot modes -- dot, point, line, and dash -- are illustrated. The four symbols illustrated are diamond, plus sign, triangle,



REGRESSION: Y=0.4+0.713333333333X. CORR: 0.93291

Figure 7.2. An example of x-y graphs with the linear regression option invoked. The vertical line segments through the diamond symbols may represent standard deviations or ranges.

- 2) Optional labelling of x and y axes (Rotation of y axis label by 90°, such as in Figure 7.1, is optional for plotter outputs.)
- 3) Optional linear regression line to be drawn on the graph as well as listing of fitted linear regression parameters and estimated correlation coefficient (See Figure 7.2.)
- 4) Inclusion of one additional symbol: circle (See item 5 of the "Original Capabilities" subsection.)
- 5) Optional standard deviation line for each plotted point (See Figure 7.2.)
- 6) Inclusion of two additional plot modes: dash and dot (See item 2 of the "Original Capabilities" subsection.)
- 7) Ability to plot more than one set of points per output (Different symbols and plot modes may be assigned for each set of points.)
- 8) A maximum of 300 points may be plotted (Previously, it was 100 points.)
- 9) Ability to change the y coordinates only while leaving the x coordinates unaltered.

Dot and Dash Modes

This subsection describes the technique used for producing the dot and dash modes. The method involves the determinations of 1) the distance between two end points of each segment, and 2) the direction or slope of each segment. Both can be easily accomplished through 1) the Euclidean distance formula, and 2) elementary trigonometry, respectively. The difference between the dot and the dash modes is that the connections of every other pairs of dots result in dashes.

The application of appropriate factors on the x and y axes is vital to the routine. For example, if the ratio between the x and y scale is 10 to 1, then the distances as computed by the Euclidean distance formula between endpoints of line segments parallel to the x axis would be ten times greater than those parallel to the y axis, given that the segments have equal lengths as measured by ruler. The result, if not remedied, would be dense dots and dashes on horizontally oriented segments as contrasted with sparse dots and dashes on vertically oriented segments.

The variable DØ is assigned as the distance between successive dots. It is one one-hundredth of the distance of the y range. What is needed is a proper transformation of DØ that is dependent on the angular orientation of the line segment such that identically dense dots and dashes among line segments are acquired regardless of their orientation. It can be shown that the transformation required is the multiplication of DØ by the factor: $D1/(((X(U1)-X(U1-1))*((T(17)-T(16))/T(7)-T(6))))^{1/2}+(Y(U1)-Y(U1-1))^{1/2})^{1/0.5}$. This result is expressed and applied in lines 4070 through 4077 of the program listed in Appendix F. The elements (X(U1),Y(U1)) and (X(U1-1),Y(U1-1)) are the respective coordinates of two successive points. The variable D1 is the distance between two successive points. The expressions T(17)-T(16) and T(7)-T(6) are the distances of the y and x ranges, respectively.

8. TEXT PLOT

This program allows text materials, especially those involving Greek alphabets and mathematical symbols, to be plotted and saved on tape. The following is the program's UDK function list:

- 1) Begin data entry
- 2) Stop data entry
- 3) Resume data entry
- 4) Print first page of text
- 5) Print next page of text
- 6) Edit text
- 7) Set character sizes
- 8) Plot first page of text
- 9) Plot next page of text
- 10) Save text.

Data Entry

The UDK's 1, 2, and 3 are self explanatory. Data may be entered via keyboard, tape, or both. The '#', '@', '^', and '&' symbols are special reserved characters used for controlling final plotted outputs. (Figure 8.2 represents a final plotted output.) The '#' entry by the user marks the end of a page. The two symbols '@' and '^' delimits subscripts and superscripts, respectively. Used as a prefix, the symbol '&' changes the next immediately following letter, or the suffix, into a Greek alphabet or a mathematical symbol. Table 8.1 is the list of conversions that '&' can accomplish. If any suffix characters other than those listed on Table

Table 8.1. Functions of the reserved character '&'.

8.1 is appended to '&', then that letter remains unchanged and '&' is eliminated from the plotted text. For example, if '&@' is entered, then '@' will be plotted because '@' is not a suffix symbol. The entry '&&' yields '&' because the second '&' is not allowed as a suffix, the first '&' is ignored, and the second '&' is not converted and plotted as is.

Other UDK Functions

The UDK's 4 and 5 are used for displaying texts on the screen. The displayed texts are identical to the entered text except for 1) any subsequent editorial changes, and 2) the addition of the reserved character '#' used for controlling line feeds and paging. Figure 8.1 is an example of the results of UDK's 4 and 5.

The UDK 6 allows text materials to be edited. It prompts the user for the character string being considered for editing and then searches for it. If the string is not unique in the text, only the first one encountered will be edited. If the string exists and is found, the user will be prompted to enter the replacement character string. Otherwise, the user will be informed that the string is not found. The user may wish to verify that the editing function is accomplished by using the UDK's 4 and 5 before plotting.

The UDK 7 allows character sizes to be assigned. The user selects one size for the subscripts and/or superscripts, and another for the main text. The program automatically calculates the largest character size that could allow a page of text to be fitted on the plotter. (If the user-assigned character sizes are too large, there may not be enough space on the plotter to hold an entire page of text.)

For the decision of selecting \hat{p}_i as the largest among p_1, p_2, \dots, p_k , the loss is $L(p^*) = p^* - p_i$. Let us suppose the posterior distribution is a Dirichlet distribution with parameters $\alpha_1, \alpha_2, \dots, \alpha_k$. We like to select \hat{p}_i such that the Bayes risk, $E L(p^*, \hat{p}_i) = \sum_j \alpha_j p^* dF(j) - \alpha_i / \sum_{i=1}^k \alpha_i$, is minimized.

Clearly, $E L(p^*, \hat{p}_i)$ is minimized when $\alpha_i = \max(\alpha_1, \alpha_2, \dots, \alpha_k)$. Therefore, the Bayes decision is to select \hat{p}_i corresponding to α_i which is maximum among $\alpha_1, \alpha_2, \dots, \alpha_k$.

Figure 8.1. Example of the function of UDK 4. This version can be edited using UDK 6, and is the material being stored in the memory. The entered text is similar to this except for editorial changes and the carriage returns in place of the '#' characters here. See text for an explanation on the usages of the reserved characters '#', '@', '^', and '&'. (The character '^' is displayed as '^' in this example.) Figure 8.2 is the associated final plot.

For the decision of selecting π_i^* as the largest among $\pi_1, \pi_2, \dots, \pi_k$, the loss is $L(\pi) = \pi^* - \pi_i$. Let us suppose the posterior distribution is a Dirichlet distribution with parameters $\alpha_1', \alpha_2', \dots, \alpha_k'$. We like to select π_i^* such that the Bayes risk, $EL(\pi^*, \pi_i) = \int_{\Theta} \pi^* dF(\theta) - \alpha_i'/\sum_{i=1}^k \alpha_i'$, is minimized.

Clearly, $EL(\pi^*, \pi_i)$ is minimized when $\alpha_i' = \max(\alpha_1', \alpha_2', \dots, \alpha_k')$. Therefore, the Bayes decision is to select π_i^* corresponding to α_i' which is maximum among $\alpha_1', \alpha_2', \dots, \alpha_k'$.

Figure 8.2. Example of a plotted output. This plot is produced when UDK 8 is pressed after the text material in Figure 8.1 is entered.

The UDK's 8 and 9 are used for producing the final output plot on the plotter or terminal screen. Figure 8.2 is an example of the result of UDK's 8 and 9. It is produced after a text similar to that in Figure 8.1 is entered. Together, Figures 8.1 and 8.2 illustrate some of the usages of the reserved characters.

The UDK 10 saves the input text on tape for subsequent retrieval and plotting.

9. BLOCK PLOT

This program plots entered texts as block letters for visual display purposes.

Input Variables

Table 9.1 is a list of the input variables with their corresponding variable names.

Thickness of Lines

The block letter effect is created by retracing the letter several times. Each time that the letter is traced, it is a small distance off from the previous tracings. This offset is done in a circular manner. The diameter of this circular path is the thickness of the lines of the block letters. Figure 9.1 demonstrates how this technique may result in the block letter H. If the circular path of Figure 9.1 were smaller, and the lines were drawn closer to each other, then a solid block letter would be the result.

The diameter of the circular path or the thickness of lines is measured according to the margins set on the plotter. The total distance between the horizontal margins measures 150 units. Figure 9.2 is an example of an output produced by using 2 units for the thickness of lines.

Character Size

The distance between the vertical margins measures 100 units. Therefore, the maximum height of the characters is 100 units divided by the

Table 9.1. List of input variables: programmer-assigned variable names and their corresponding descriptions.

<u>Programmer-assigned Name</u>	<u>Variable Description</u>
Z\$	Text to be plotted
W	Width or thickness of lines of plotted characters
H	Height of plotted characters

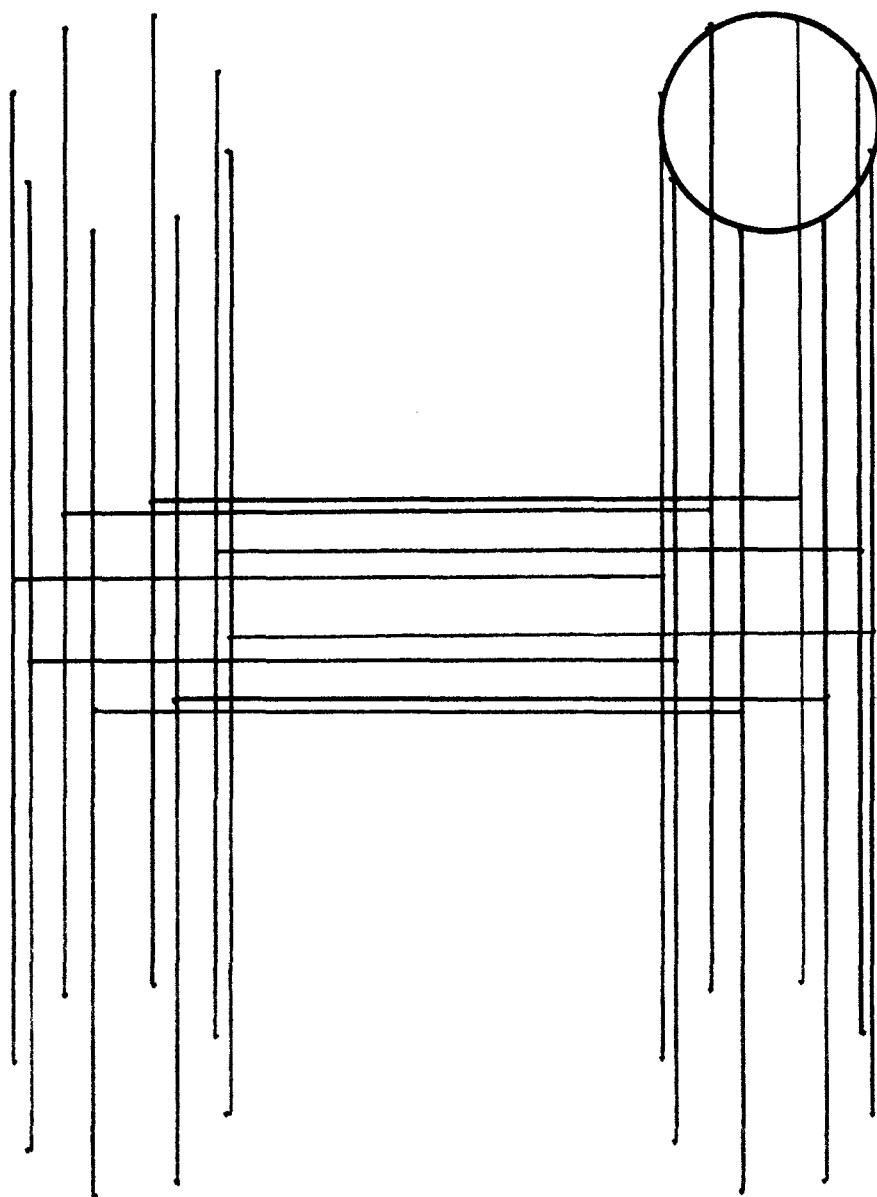


Figure 9.1. A demonstration of the mechanism used to produce block letters.
See the "Thickness of Lines" subsection of text for explanation.

SMOKING

FORBIDDEN

Figure 9.2. An example of "Block Plot" outputs. The user has assigned two units for the thickness of lines and 49 units for the height of characters to produce this output.

number of lines in the text. The actual character height is about 2/3 of the input height to allow some spacing between the lines. The user-input height of the characters in Figure 9.2 is 49 units. If a smaller height were assigned, the characters would be shorter, and there would be more space between the lines.

The character width is automatically calculated by the program such that the entire horizontal margin range is filled by the longest lines of text.

Appendix A -- Pie-chart Program Listing

```
100 PAGE
110 INIT
120 I6=0
125 I1=0
126 H$=CHR(8)
127 J$=CHR(10)
128 K$=CHR(11)
129 G$=CHR(7)
130 DIM A(50),B(6),Q$(300),C(6),D(6),S$(60),R$(100)
140 DIM E(6),F(6),G(6),L(15),H(6),M(15)
150 SET DEGREES
160 PRINT "YOU HAVE JUST INITIATED A PIE-CHART PLOTTING PROGRAM."
170 PRINT "DO YOU WISH TO PLOT ON PAPER? (N,Y) ";
180 INPUT B$
190 IF B$="N" THEN 520
200 PRINT "LOAD YOUR PAPER ON PLOTTER."
210 PRINT "BE SURE THAT LOWER LEFT AND UPPER RIGHT ARE AT THE EXTREME"
220 PRINT "CORNERS OF TABLET."
230 PRINT "IF NOT, RESET MARGINS BY TURNING PLOTTER OFF AND THEN ON."
240 PRINT "MAKE SURE THAT ALL WHITE BUTTONS ON THE PLOTTER ARE IN "
250 PRINT "THE UP POSITION."
260 PRINT "PRESS RETURN WHEN YOU ARE READY TO GO ON."
270 INPUT A$
280 PRINT "STATION PEN AT THE LOWER LEFT CORNER OF YOUR PAPER"
290 PRINT "USING THE JOYSTICK."
300 PRINT "PRESS RETURN WHEN READY."
310 INPUT A$
320 GIN @1:X1,Y1
330 PRINT "STATION PEN AT THE UPPER RIGHT CORNER OF YOUR PAPER"
340 PRINT "USING THE JOYSTICK."
350 PRINT "PRESS RETURN WHEN READY."
360 INPUT A$
370 GIN @1:X2,Y2
380 IF X1=>X2 THEN 410
390 IF Y1=>Y2 THEN 410
400 GO TO 440
410 PAGE
420 PRINT G$;"TRY AGAIN."
430 GO TO 280
440 X3=X1/2+10
450 X4=X2/2+10
460 Y3=Y1/2+40
470 Y4=Y2/2+40
480 GOSUB 2470
490 PRINT "PRESS RETURN TO GO ON."
500 INPUT A$
510 PAGE
515 IF I1=4 THEN 800
```

```
520 GOSUB 2390
530 PRINT " "
540 PRINT "HOW MANY PIES DO YOU WANT TO PLOT? ";
550 INPUT T
560 IF T>0 AND T<7 THEN 590
570 PRINT "WE CAN ONLY PLOT FROM 1 TO 6 PIES."
580 GO TO 540
590 PRINT "INPUT CENTER AND RADII OF PIE CHARTS."
600 FOR I=1 TO T
610 PRINT "X LOCATION OF PIE #";I;"..... ";
620 INPUT C(I)
630 IF C(I)>0 AND C(I)<150 THEN 660
640 PRINT "CENTER OF X LOCATION CAN BE BETWEEN 0 AND 150 ONLY."
650 GO TO 610
660 PRINT "Y LOCATION OF PIE #";I;"..... ";
670 INPUT D(I)
680 IF D(I)>0 AND D(I)<100 THEN 710
690 PRINT "CENTER OF Y LOCATION CAN BE BETWEEN 0 AND 100 ONLY."
700 GO TO 660
710 PRINT "RADIUS OF PIE #";I;"..... ";
720 INPUT B(I)
730 IF B(I)>0 AND B(I)<50 THEN 760
740 PRINT "LENGTH OF RADIUS CAN BE BETWEEN 0 AND 50 ONLY."
750 GO TO 710
760 F(I)=D(I)/2+40
770 E(I)=C(I)/2+10
780 G(I)=B(I)/2
790 NEXT I
800 GOSUB 2470
810 GOSUB 2790
820 PRINT "DO YOU WANT TO MAKE ANY CHANGES ON THE POSITION? (N,Y) ";
830 INPUT A$
840 IF A$="Y" THEN 510
850 IF I6=0 THEN 890
860 PRINT "DO YOU NEED TO ENTER THE PERCENTAGE DATA? ";G$;"(N,Y)...";
870 INPUT A$
880 IF A$="N" THEN 1300
890 PAGE
900 PRINT "INPUT THE NUMBER OF PORTIONS ON PIE CHART. ";
910 INPUT P
920 IF P>0 AND P<50/T THEN 950
930 PRINT "LIMIT EXCEEDED. TRY AGAIN.";G$
940 GO TO 900
950 PRINT " "
960 PRINT "THERE ARE FIVE POSSIBLE SHADES FOR HIGHLIGHT:"
970 PRINT "1) CLEAR."
980 PRINT "2) LIGHT GREY."
990 PRINT "3) MEDIUM GREY."
1000 PRINT "4) DARK GREY."
1010 PRINT "5) EXTRA DARK GREY."
```

```

1020 PRINT " "
1030 PRINT "ENTER THE SHADE NUMBER FOR EACH PORTION."
1040 FOR I=1 TO P
1050 PRINT "PORTION #";I;"..... ";
1060 INPUT M(I)
1070 IF M(I)>1 AND M(I)<=5 THEN 1100
1080 PRINT G$;"LIMIT EXCEEDED."
1090 GO TO 1050
1100 NEXT I
1110 PRINT " "
1120 PRINT "INPUT PERCENTAGE OF EACH PORTION OF EACH PIE."
1130 FOR J=1 TO T
1135 PRINT " "
1140 RO=0
1150 FOR I=1 TO P
1160 K=(J-1)*P+I
1170 PRINT "PIE #";J; ", PORTION #";I;"..... ";
1180 INPUT A(K)
1190 RO=RO+A(K)
1200 IF A(K)>0 AND A(K)<=100 AND RO<=100 THEN 1230
1210 PRINT G$;"LIMIT EXCEEDED."
1220 GO TO 1140
1230 A(K)=A(K)*3.6
1240 NEXT I
1250 IF RO>100 THEN 1290
1260 PRINT G$;"PORTION PERCENTAGES WITHIN A PIE MUST ADD UP TO 100."
1270 I=I-1
1280 GO TO 1140
1290 NEXT J
1300 PAGE
1310 IF I6=0 THEN 1348
1320 PRINT "DO YOU NEED TO CHANGE LABELS? (N,Y)";G$;
1330 INPUT A$
1340 IF A$="N" THEN 1600
1348 J2=0
1350 PRINT "ENTER GRAPH LABEL..... ";
1360 INPUT S$
1370 PRINT "ENTER CHARACTER SIZE FOR GRAPH LABEL. (WIDTH,HEIGHT)... ";
1380 INPUT Z1,Z2
1382 PRINT "DO YOU WISH TO UNDERLINE GRAPH LABEL? (N,Y)... ";
1384 INPUT A$
1386 IF A$="N" THEN 1390
1388 J2=J2+1
1390 R$=" "
1400 PRINT " "
1410 FOR J=1 TO T
1420 PRINT "ENTER LABEL FOR PIE #";J;"..... ";
1430 INPUT A$
1440 H(J)=LEN(A$)
1450 R$=R$&A$

```

```

1460 NEXT J
1470 PRINT "ENTER CHARACTER SIZE FOR PIE LABEL. (WIDTH,HEIGHT)... ";
1480 INPUT Z3,Z4
1482 PRINT "DO YOU WISH TO UNDERLINE PIE LABEL? (N,Y)... ";
1484 INPUT A$
1486 IF A$="N" THEN 1500
1488 J2=J2+2
1500 PRINT " "
1505 Q$=" "
1510 PRINT "ENTER LABEL FOR EACH PORTION OF PIE."
1520 FOR I=1 TO P
1530 PRINT "ENTER LABEL FOR PORTION #";I;"..... ";
1540 INPUT A$
1550 L(I)=LEN(A$)
1560 Q$=Q$&A$
1570 NEXT I
1580 PRINT "ENTER CHARACTER SIZE FOR PORTION LABEL. (WIDTH,HEIGHT)... ";
1590 INPUT Z5,Z6
1600 PRINT " "
1610 PRINT "WE ARE NOW READY TO DISPLAY OUTPUT."
1620 IF B$="Y" THEN 1670
1630 O=32
1640 PRINT "PRESS RETURN TO GO ON."
1650 INPUT A$
1660 GO TO 1700
1670 PRINT "WHERE DO YOU WANT YOUR OUTPUT TO GO?"
1680 PRINT "ENTER 1 FOR PLOTTER, 32 FOR SCREEN..... ";
1690 INPUT O
1700 PAGE
1710 I2=2
1720 I3=0
1730 I4=150
1740 I5=0
1750 FOR J=1 TO T
1760 R1=0
1770 R9=2
1780 MOVE @0:B(J)*SIN(O)+C(J),B(J)*COS(O)+D(J)
1790 FOR K=0 TO 360 STEP 2
1800 DRAW @0:B(J)*SIN(K)+C(J),B(J)*COS(K)+D(J)
1810 NEXT K
1820 PRINT @0,17:Z3,Z4
1830 MOVE @0:C(J),D(J)+B(J)
1840 PRINT @0:K$:K$;
1845 IF J2<2 THEN 1924
1850 FOR I1=1 TO H(J) STEP 2
1860 PRINT @0:" ";
1870 NEXT I1
1880 PRINT @0:J$;
1890 FOR I1=1 TO H(J)
1900 PRINT @0:H$;H$;"-";

```

```

1910 NEXT I1
1920 PRINT @0:H$;K$;
1922 GO TO 1930
1924 FOR I1=1 TO H(J) STEP 2
1925 PRINT @0:H$;
1926 NEXT I1
1930 A$=SEG(R$,I2,H(J))
1940 I2=I2+H(J)
1950 PRINT @0:A$;
1960 FOR I=1 TO P
1970 I7=R1
1980 R1=R1+A((J-1)*P+I)
1990 MOVE @0:B(J)*SIN(R1)+C(J),B(J)*COS(R1)+D(J)
2000 DRAW @0:C(J),D(J)
2010 GOSUB 2920
2020 NEXT I
2030 I3=I3 MAX D(J)+B(J)
2040 I4=I4 MIN C(J)-B(J)
2050 I5=I5 MAX C(J)+B(J)
2060 NEXT J
2070 PRINT @0,17:Z1,Z2
2080 MOVE @0:(I4+I5)/2,I3
2090 PRINT @0:K$,K$,K$,K$,K$;
2092 IF J2=0 OR J2=2 THEN 2174
2100 FOR I1=1 TO LEN(S$) STEP 2
2110 PRINT @0:" ";
2120 NEXT I1
2130 PRINT @0:J$;
2140 FOR I1=1 TO LEN(S$)
2150 PRINT @0:H$;H$;"-";
2160 NEXT I1
2170 PRINT @0:H$;K$;
2172 GO TO 2180
2174 FOR I1=1 TO LEN(S$) STEP 2
2175 PRINT @0:H$;
2176 NEXT I1
2180 PRINT @0:S$;
2190 IF B$="N" THEN 2280
2200 IF O=1 THEN 2280
2210 MOVE X1,Y1
2220 DRAW X1,Y1
2230 DRAW X1,Y2
2240 DRAW X2,Y2
2250 DRAW X2,Y1
2260 DRAW X1,Y1
2270 I6=1
2280 HOME
2290 I6=1
2300 PRINT "ENTER CORRESPONDING # OF FOLLOWING: "
2310 PRINT "1) PLOT SAME GRAPH AGAIN."

```

```
2320 PRINT "2) MAKE CHANGES ON PRESENT GRAPH."
2330 PRINT "3) START AGAIN FROM BEGINNING."
2335 PRINT "4) RESTORE PLOTTER OUTPUT OPTION OR PAPER LOCATION CHANGE."
2340 PRINT "5) STOP.",G$
2350 INPUT I1
2355 PAGE
2360 GO TO I1 OF 1620,800,100,170
2380 END
2390 PRINT "SCREEN MEASURES UNIT 1 TO 130 FROM LEFT TO RIGHT."
2400 IF B$="Y" THEN 2430
2410 A$=" "
2420 GO TO 2450
2430 A$="BOTH"
2440 PRINT "PLOTTER MEASURES UNIT 1 TO 150 FROM LEFT TO RIGHT."
2450 PRINT A$;" UNIT 1 TO 100 FROM BOTTOM TO TOP."
2460 RETURN
2470 PAGE
2480 MOVE 10,40
2490 PRINT J$;"0";H$;H$;K$;"0";
2500 MOVE 10,40
2510 DRAW 75,40
2520 PRINT J$;H$;H$;H$;"130";
2530 MOVE 75,40
2540 DRAW 75,90
2550 DRAW 10,90
2560 PRINT H$;H$;H$;J$;"100";
2570 MOVE 10,90
2580 DRAW 10,40
2582 IF B$="N" THEN 2740
2590 MOVE 75,40
2591 DRAW 85,40
2592 DRAW 85,90
2593 DRAW 75,90
2594 MOVE 85,40
2600 PRINT J$;H$;H$;H$;"150";
2640 MOVE X3,Y3
2650 DRAW X3,Y4
2660 DRAW X4,Y4
2670 DRAW X4,Y3
2680 DRAW X3,Y3
2690 DRAW X4,Y4
2700 MOVE X4,Y3
2710 DRAW X3,Y4
2720 MOVE 35,92
2730 PRINT "PLOTTER"
2740 MOVE 2,30
2750 PRINT "ENCLOSING RECTANGLE IS THE OUTLINE OF PLOTING SURFACE."
2760 IF B$="N" THEN 2780
2770 PRINT "CROSSED AREA IS YOUR PAPER."
2780 RETURN
```

```

2790 FOR J=1 TO T
2800 MOVE E(J),F(J)
2810 FOR I=1 TO 100
2820 DRAW E(J),F(J)
2830 NEXT I
2840 MOVE G(J)*SIN(O)+E(J),G(J)*COS(O)+F(J)
2850 FOR I=1 TO 370 STEP 10
2860 DRAW G(J)*SIN(I)+E(J),G(J)*COS(I)+F(J)
2870 NEXT I
2880 NEXT J
2890 MOVE 2,22
2900 PRINT "THE CIRCLES INDICATE POSITIONS OF PIE CHART."
2910 RETURN
2920 R2=B(J)*(M(I)-1)
2930 IF R2=0 THEN 3130
2940 R3=B(J)*2/R2
2950 FOR I1=0 TO R2
2960 FOR J1=0 TO R2
2970 R4=B(J)-(I1+0.1)*R3
2980 R5=B(J)-(J1+0.1)*R3
2990 IF R4^2+R5^2>B(J)^2 THEN 3110
3000 R8=ATN(ABS(R4/R5))
3010 I9=R8
3020 I8=(R4>0)+2*(R5>0)+1
3030 GO TO I8 OF 3070,3040,3050,3060
3040 R8=R8-180
3050 R8=-R8
3060 I2=180+R8
3070 IF I9>I7 AND I9<R1 THEN 3090
3080 GO TO 3110
3090 MOVE @0:C(J)-R4,D(J)-R5
3100 DRAW @0:C(J)-R4,D(J)-R5
3110 NEXT J1
3120 NEXT I1
3130 R3=R1-0.5*A((J-1)*P+I)
3140 PRINT @0,17:Z5,Z6
3150 MOVE @0:B(J)*SIN(R3)+C(J),B(J)*COS(R3)+D(J)
3160 A$=SEG(Q$,R9,L(I))
3170 R9=R9+L(I)
3180 R3=INT(R3/90)+1
3190 GO TO R3 OF 3200,3200,3220,3240
3200 RMOVE @0:1,0
3210 IF R3=1 THEN 3280
3220 PRINT @0:J$;
3230 IF R3=2 THEN 3280
3240 FOR I1=1 TO L(I)
3250 PRINT @0:H$;
3260 NEXT I1
3270 GO TO 3280
3280 PRINT @0:A$;

```

3290 RETURN

Appendix B -- 3-d Transformation Program Listing

```
1 INIT
2 DIM Q$(201),N$(201)
3 GO TO 100
4 GO TO 430
8 GO TO 510
12 GO TO 590
16 GO TO 670
20 GO TO 710
24 GO TO 750
28 GO TO 790
32 GO TO 830
36 GO TO 890
40 GO TO 2550
44 GO TO 2470
48 GO TO 950
52 GO TO 3100
56 GO TO 3170
60 B7=1
61 GO TO 2340
64 B7=2
65 GO TO 2340
68 GO TO 3610
72 D9=2
73 GO TO 1850
76 D9=1
77 GO TO 1850
80 GO TO 280
100 Q$=" "
105 G9=1
110 J3=5
120 C1=2
130 I=1
140 D4=2
150 D1=0
160 D2=0
170 D3=0
180 T8=1.2
190 T9=1.2
200 T0=1.2
210 N$=" "
220 J7=1
230 O=32
240 J8=1
250 SET DEGREES
260 PRINT "PRESS UDK #12 TO INPUT DATA, UDK #13 TO INPUT FUNCTION."
270 DIM X(200),Y(200),Z(200),D(200),E(200),F(200)
272 END
280 PAGE
```

```
290 PRINT " "
300 PRINT "ENTER THE # OF THE FOLLOWING ITEMS :"
310 PRINT "1) SET DEGREE OF EACH ROTATION"
320 PRINT "2) SET CENTER POINT OF ROTATION"
330 PRINT "3) SET TRANSLATION UNITS"
340 PRINT "4) SET SCALING PARAMETERS"
350 PRINT "5) SET DISPLAY OF CENTER OF ROTATION"
360 PRINT "6) SET DISPLAY OF ROTATED ANGLES"
370 PRINT "7) SET OUTPUT DEVICE"
380 PRINT "8) SET RETENTION OF ALL TRANSFORMATIONS ON SCREEN"
390 INPUT B8
400 GO TO B8 OF 1300,2090,3510,3560,2130,2300,2170,3800
410 PRINT " "
420 END
430 FOR J=1 TO I-1
440 T1=D(J)
450 T2=F(J)
460 D(J)=(T1-H)*COS(J3)-(T2-L)*SIN(J3)+H
470 F(J)=(T1-H)*SIN(J3)+(T2-L)*COS(J3)+L
480 NEXT J
490 D1=D1+J3
500 GO TO 2550
510 FOR J=1 TO I-1
520 T1=E(J)
530 T2=D(J)
540 D(J)=(T1-K)*SIN(J3)+(T2-H)*COS(J3)+H
550 E(J)=(T1-K)*COS(J3)-(T2-H)*SIN(J3)+K
560 NEXT J
570 D2=D2+J3
580 GO TO 2550
590 FOR J=1 TO I-1
600 T1=E(J)
610 T2=F(J)
620 E(J)=(T2-L)*SIN(J3)+(T1-K)*COS(J3)+K
630 F(J)=(T2-L)*COS(J3)-(T1-K)*SIN(J3)+L
640 NEXT J
650 D3=D3+J3
660 GO TO 2550
670 FOR J=1 TO I-1
680 F(J)=F(J)+T7
690 NEXT J
700 GO TO 2550
710 FOR J=1 TO I-1
720 F(J)=F(J)-T7
730 NEXT J
740 GO TO 2550
750 FOR J=1 TO I-1
760 D(J)=D(J)-T6
770 NEXT J
780 GO TO 2550
```

```
790 FOR J=1 TO I-1
800 D(J)=D(J)+T6
810 NEXT J
820 GO TO 2550
830 FOR J=1 TO I-1
840 D(J)=D(J)*T8
850 E(J)=E(J)*T9
860 F(J)=F(J)*T0
870 NEXT J
880 GO TO 2550
890 FOR J=1 TO I-1
900 D(J)=D(J)/T8
910 E(J)=E(J)/T9
920 F(J)=F(J)/T0
930 NEXT J
940 GO TO 2550
950 PAGE
960 PRINT "ENTER DATA VIA KEYBOARD? ";
970 GOSUB 3040
980 IF U0=2 THEN 1340
990 PRINT "INSERT TAPE AND ENTER FILE NUMBER"
1000 INPUT T3
1010 FIND T3
1020 READ @33:T4
1030 IF I<=1 THEN 1080
1040 PRINT "DO YOU WISH TO ADD TO PREVIOUS DATA (N OR Y)?";
1050 GOSUB 3040
1060 J7=U0
1070 GO TO U0 OF 1080,1110
1080 I1=1
1090 FO=0
1100 GO TO 1120
1110 I1=I
1120 I2=I1+T4-1
1130 IF I2>=380 THEN 1160
1140 PRINT "TOO MANY DATA POINTS. LIMIT IS 380."
1150 END
1160 FOR I=I1 TO I2
1170 READ @33:X(I),Y(I),Z(I)
1180 D(I)=X(I)
1190 E(I)=Y(I)
1200 F(I)=Z(I)
1210 NEXT I
1220 READ @33:N$
1230 GO TO J7 OF 1280,1240
1240 N$=SEG(N$,2,T4)
1250 Q$=Q$&N$
1260 N$=" "
1270 GO TO 1620
1280 Q$=N$
```

```
1290 GO TO 1620
1300 PRINT "THE PRESENT NUMBER OF DEGREES PER ROTATION IS ";J3
1310 PRINT "ENTER NEW DEGREES"
1320 INPUT J3
1330 END
1340 IF I<=1 THEN 1380
1350 PRINT "DO YOU WISH TO ADD TO PREVIOUS DATA (N OR Y)?";
1360 GOSUB 3040
1370 GO TO U0 OF 1380,1420
1380 I1=1
1390 Q$=""
1400 F0=0
1410 GO TO 1430
1420 I1=I
1430 PRINT "TYPE IN YOUR DATA, M TO DISCONTINUE."
1440 FOR I=I1 TO 380
1450 PRINT "X(";I;")....";
1460 GOSUB 3000
1470 X(I)=VAL(U$)
1480 PRINT "Y(";I;")....";
1490 GOSUB 3000
1500 Y(I)=VAL(U$)
1510 PRINT "Z(";I;")....";
1520 GOSUB 3000
1530 Z(I)=VAL(U$)
1540 PRINT "CONNECTED TO PREVIOUS POINT (N OR Y)?";
1550 INPUT P$
1560 P$=SEG(P$,1,1)
1570 Q$=Q$&P$
1580 D(I)=X(I)
1590 E(I)=Y(I)
1600 F(I)=Z(I)
1610 NEXT I
1620 H3=X(1)
1630 H4=X(1)
1640 H5=Y(1)
1650 H6=Y(1)
1660 H7=Z(1)
1670 H8=Z(1)
1680 FOR J=2 TO I-1
1690 H3=H3 MIN X(J)
1700 H4=H4 MAX X(J)
1710 H5=H5 MIN Y(J)
1720 H6=H6 MAX Y(J)
1730 H7=H7 MIN Z(J)
1740 H8=H8 MAX Z(J)
1750 NEXT J
1760 G=H4 MAX H6 MAX H8
1770 H1=H3 MIN H5 MIN H7
1780 WINDOW 2.45*H1-1.45*G,2.45*G-1.45*H1,2*H1-G,2*G-H1
```

```
1790 H=(H3+H4)/2
1800 K=(H5+H6)/2
1810 L=(H7+H8)/2
1820 T6=(H4-H3)/5
1830 T7=(H8-H7)/5
1840 END
1850 PRINT "INSERT TAPE AND ENTER FILE NUMBER";
1860 INPUT T3
1870 FIND T3
1880 PRINT "THERE IS A TOTAL OF ";I-1;" POINTS."
1890 PRINT "ENTER THE FIRST AND LAST ITEM TO BE SAVED.";
1900 INPUT T4,T5
1910 IF T4>1 AND T4<=I-1 THEN 1930
1920 GO TO 1880
1930 IF T5>1 AND T5<=I-1 THEN 1950
1940 GO TO 1880
1950 T6=T5-T4+1
1960 WRITE T6
1970 FOR J=T4 TO T5
1980 GO TO D9 OF 1990,2010
1990 WRITE D(J),E(J),F(J)
2000 GO TO 2020
2010 WRITE X(J),Y(J),Z(J)
2020 NEXT J
2030 N$=SEG(Q$,T4+1,T6)
2040 N$=" "&N$
2050 WRITE N$
2060 N$=" "
2070 CLOSE
2080 END
2090 PRINT "CURRENT ROTATION POINT IS ( ";H,K,L;" )."
2100 PRINT "ENTER NEW ROTATION POINT ";
2110 INPUT H,K,L
2120 END
2130 PRINT "DO YOU WANT TO DISPLAY POINT OF ROTATION (N OR Y)?"
2140 GOSUB 3040
2150 C1=U0
2160 END
2170 IF O=1 THEN 2240
2180 PRINT "CURRENT OUTPUT DEVICE IS SCREEN"
2190 PRINT "DO YOU WANT TO CHANGE THE OUTPUT DEVICE TO PLOTTER?"; 
2200 GOSUB 3040
2210 GO TO U0 OF 2230,2220
2220 O=1
2230 END
2240 PRINT "CURRENT OUTPUT DEVICE IS PLOTTER"
2250 PRINT "DO YOU WANT TO CHANGE THE OUTPUT DEVICE TO SCREEN?"; 
2260 GOSUB 3040
2270 GO TO U0 OF 2290,2280
2280 O=32
```

```
2290 END
2300 PRINT "DO YOU WANT TO DISPLAY ROTATED ANGLE (N OR Y)?"
2310 GOSUB 3040
2320 D4=U0
2330 END
2340 PAGE
2350 PRINT @0:"ITEM", "X COORDINATE", "Y COORDINATE", "Z COORDINATE"
2360 PRINT @0:"CONNECTION"
2370 PRINT @0:"-----"
2380 PRINT @0: " "
2390 FOR J=1 TO I-1
2400 V$=SEG(Q$,J+1,1)
2410 IF B7=2 THEN 2440
2420 PRINT @0:J,X(J),Y(J),Z(J),V$
2430 GO TO 2450
2440 PRINT @0:J,D(J),E(J),F(J),V$
2450 NEXT J
2460 END
2470 FOR J=1 TO I-1
2480 D(J)=X(J)
2490 E(J)=Y(J)
2500 F(J)=Z(J)
2510 NEXT J
2520 D1=0
2530 D2=0
2540 D3=0
2550 IF J8=2 THEN 2570
2560 PAGE
2570 IF C1=1 THEN 2610
2580 MOVE @0:H,L
2590 RMOVE @0:(3.9*H1-3.9*G)*0.0043,(3*H1-3*G)*0.0087
2600 PRINT @0:"o"
2610 FOR J=1 TO I-1
2620 R$=SEG(Q$,J+1,1)
2625 IF G9=2 AND F0=1 THEN 2672
2630 IF R$="N" THEN 2660
2640 DRAW @0:D(J),F(J)
2650 GO TO 2680
2660 MOVE @0:D(J),F(J)
2670 RDRAW @0:0,0
2671 GO TO 2680
2672 IF R$="N" THEN 2680
2673 G6=E(J-1)
2674 G7=F(J-1)
2675 G5=D(J-1)
2676 G8=1
2677 GOSUB 6000
2680 NEXT J
2690 IF F0=0 THEN 2820
2700 MOVE @0:D(1),F(1)
```

```

2710 FOR J0=1 TO X4*Y4
2730 K1=(J0-1)*X4-(Y4*X4-1)*INT((J0-1)/Y4)+2
2734 J=K1-1
2740 R$=SEG(Q$,K1,1)
2750 N$=SEG(Q$,K1-X4,1)
2755 IF G9=2 THEN 2791
2760 IF N$=R$ THEN 2787
2770 MOVE @0:D(J),F(J)
2780 GO TO 2810
2787 DRAW @0:D(J),F(J)
2790 GO TO 2810
2791 IF N$>R$ THEN 2810
2792 K1=J-X4
2793 G5=D(K1)
2794 G6=E(K1)
2795 G7=F(K1)
2796 G8=2
2797 GOSUB 6000
2810 NEXT J0
2820 IF D4=1 THEN 2990
2830 D5=(D1+0.5)/360
2840 D6=(D2+0.5)/360
2850 D7=(D3+0.5)/360
2860 W1=INT(D5)
2870 W1=(D5-W1)*360
2880 W3=INT(D7)
2890 W2=INT(D6)
2900 W2=(D6-W2)*360
2910 W3=(D7-W3)*360
2920 W1=INT(W1)
2930 W2=INT(W2)
2940 W3=INT(W3)
2950 HOME
2960 PRINT "                                ROTATED ANGLE"
2970 PRINT "                                C-CLOCKWISE VERT-AXIS HORI-AXIS"
2980 PRINT "                                ";W1;"           ";W2,W3
2990 END
3000 INPUT U$
3010 IF U$="M" THEN 1620
3020 U$=U$&",0"
3030 RETURN
3040 INPUT R$
3050 U0=POS("NY",R$,1)
3060 IF U0=0 THEN 3040
3070 PAGE
3080 RETURN
3090 END
3100 PAGE
3105 DELETE 3150,3159
3110 PRINT "TYPE IN Z5 AS A FUNCTION OF X5 AND Y5"

```

```
3120 PRINT "START AT LINE 3150, END AT 3159."
3130 PRINT "PRESS UDK #14 WHEN YOU ARE FINISHED."
3140 END
3160 RETURN
3170 PRINT "WHAT IS MINIMUM X, MAXIMUM X, X INCREMENTATION?"
3180 INPUT X1,X3,X2
3190 PRINT "WHAT IS MINIMUM Y, MAXIMUM Y, Y INCREMENTATION?"
3200 INPUT Y1,Y3,Y2
3202 PRINT "DO YOU WANT TO REMOVE HIDDEN LINES? ";
3204 GOSUB 3040
3206 G9=U0
3210 X4=INT((X3-X1)/X2)+1
3220 Y4=INT((Y3-Y1)/Y2)+1
3230 IF X4*Y4>380 THEN 1140
3240 FOR J=1 TO Y4
3250 FOR I=1 TO X4
3260 X5=X1+(I-1)*X2
3270 Y5=Y1+(J-1)*Y2
3300 GOSUB 3150
3310 K1=(J-1)*X4+I
3320 X(K1)=X5
3330 D(K1)=X5
3340 Y(K1)=Y5
3350 E(K1)=Y5
3360 Z(K1)=Z5
3370 F(K1)=Z5
3380 NEXT I
3390 NEXT J
3392 Q$=" "
3394 F0=1
3400 FOR J=1 TO Y4
3410 N$="N"
3420 Q$=Q$&N$
3430 FOR I=1 TO X4-1
3440 N$="Y"
3450 Q$=Q$&N$
3460 NEXT I
3470 NEXT J
3480 I=X4*Y4+1
3490 GO TO 1620
3500 END
3510 PRINT "PRESENT UNITS EACH TRANSLATION ARE ";T6;" FOR LEFT/RIGHT"
3520 PRINT " AND ";T7;" FOR UP/DOWN."
3530 PRINT "ENTER NEW LEFT/RIGHT AND UP/DOWN TRANSLATION UNITS : ";
3540 INPUT T6,T7
3550 END
3560 PRINT "PRESENT SCALING PARAMETERS ARE ";T8;" HORIZONTALLY,"
3570 PRINT T9;" DEEPLY, AND ";T0;" VERTICALLY."
3580 PRI "INPUT NEW HORIZONTAL, DEEP, AND VERTICAL SCALING PARAMETERS"
3590 INPUT T8,T9,T0
```

```

3600 END
3610 PAGE
3620 B1=D(1)
3630 B2=D(1)
3640 B3=E(1)
3650 B4=E(1)
3660 B5=F(1)
3670 B6=F(1)
3680 FOR J=2 TO I-1
3690 B1=B1 MIN D(J)
3700 B2=B2 MAX D(J)
3710 B3=B3 MIN E(J)
3720 B4=B4 MAX E(J)
3730 B5=B5 MIN F(J)
3740 B6=B6 MAX F(J)
3750 NEXT J
3760 PRINT "THE MINIMUMS AND MAXIMUMS HORIZONTALLY, DEEPLY, AND"
3770 PRINT "VERTICALLY ARE RESPECTIVELY :"
3780 PRINT "(";B1;",";B2;"), (";B3;",";B4;"), (";B5;",";B6;") . "
3790 END
3800 PRINT "DO YOU WANT RETENTION OF OLD TRANFORMATIONS ON SCREEN : ";
3810 GOSUB 3040
3820 J8=U0
3830 END
6000 Y8=0
6001 G0=G5
6002 H0=G7
6003 G1=0
6005 Y0=1
6010 IF D(J)<>G5 THEN 6040
6030 GO TO 6050
6040 M1=(F(J)-G7)/(D(J)-G5)
6050 X7=D(J) MIN G5
6060 X8=D(J) MAX G5
6070 X9=D(J)
6075 X0=X9
6080 Y9=F(J)
6090 FOR I9=1 TO Y4
6100 FOR I8=1 TO X4-1
6105 J9=0
6110 K1=(I9-1)*X4+I8
6115 D0=D(K1+1)
6116 D8=D(K1+1) MIN D(K1)
6117 D9=D(K1+1) MAX D(K1)
6120 IF K1=J-2 OR K1=J AND G8=1 THEN 6200
6130 IF K1=J OR K1=J-1 OR K1=J-X4 OR K1=J-X4-1 AND G8=2 THEN 6200
6140 IF D(K1+1)<>D(K1) THEN 6170
6150 J9=1
6160 GO TO 6180
6170 M2=(F(K1+1)-F(K1))/(D0-D(K1))

```

```
6180 GOSUB 7000
6200 NEXT I8
6210 NEXT I9
6220 FOR I9=1 TO Y4-1
6230 FOR I8=1 TO X4
6235 J9=0
6240 K1=(I9-1)*X4+I8
6245 DO=D(K1+X4)
6246 D8=D(K1+X4) MIN D(K1)
6247 D9=D(K1+X4) MAX D(K1)
6250 IF K1=J-1 OR K1=J OR K1=J-X4-1 OR K1=J-X4 AND G8=1 THEN 6400
6260 IF K1=J OR K1=J-2*X4 AND G8=2 THEN 6400
6270 IF D(K1+X4)<D(K1) THEN 6300
6280 J9=1
6290 GO TO 6310
6300 M2=(F(K1+X4)-F(K1))/(DO-D(K1))
6310 GOSUB 7000
6400 NEXT I8
6410 NEXT I9
6415 Y8=Y0
6416 Y0=1
6420 Z7=(X9+G0)/2
6430 Z8=(Y9+H0)/2
6432 IF D(J)<G5 THEN 6440
6434 IF F(J)=G7 THEN 6437
6435 Z9=G6+(Z8-G7)*(E(J)-G6)/(F(J)-G7)
6436 GO TO 6450
6437 Z9=G6 MIN E(J)
6438 GO TO 6450
6440 Z9=G6+(Z7-G5)*(E(J)-G6)/(D(J)-G5)
6450 FOR I9=1 TO Y4-1
6460 FOR I8=1 TO X4-1
6462 O9=2
6465 K1=(I9-1)*X4+I8
6466 K2=(I9-1)*X4+I8+1
6467 K3=I9*X4+I8+1
6468 K4=I9*X4+I8
6470 A1=D(K1)
6471 A2=F(K1)
6472 A3=D(K2)
6473 A4=F(K2)
6474 A5=D(K3)
6475 A6=F(K3)
6476 GOSUB 8000
6477 O1=A
6480 A3=D(K4)
6481 A4=F(K4)
6482 GOSUB 8000
6483 O2=A
6485 A5=D(K2)
```

```
6486 A6=F(K2)
6487 GOSUB 8000
6488 03=A
6490 A1=D(K3)
6491 A2=F(K3)
6492 GOSUB 8000
6493 04=A
6495 09=0
6500 IF 01+02>03+04 THEN 6580
6504 IF 01+02=03+04 THEN 6510
6506 09=1
6510 A3=Z7
6511 A4=Z8
6512 GOSUB 8000
6513 05=A
6514 A1=D(K1)
6515 A2=F(K1)
6516 GOSUB 8000
6517 05=A+05
6518 A5=D(K3)
6519 A6=F(K3)
6520 GOSUB 8000
6521 05=A+05
6525 IF 01=05 THEN 6550
6530 05=A
6531 A5=D(K4)
6532 A6=F(K4)
6533 GOSUB 8000
6534 05=A+05
6540 A1=D(K3)
6541 A2=F(K3)
6542 GOSUB 8000
6543 05=A+05
6544 IF 02=05 THEN 6650
6545 GO TO 6700
6550 L1=D(K1)
6551 L2=E(K1)
6552 L3=F(K1)
6553 L4=D(K2)
6554 L5=E(K2)
6555 L6=F(K2)
6556 L7=D(K3)
6557 L8=E(K3)
6558 L9=F(K3)
6560 GOSUB 9000
6565 IF L0>Z9 THEN 6575
6570 GO TO 6800
6575 IF 09=1 THEN 6700
6580 A1=D(K4)
6581 A2=F(K4)
```

6582 A5=D(K1)
6583 A6=F(K1)
6584 GOSUB 8000
6585 O5=A
6586 A1=D(K2)
6587 A2=F(K2)
6588 GOSUB 8000
6589 O5=A+O5
6590 A5=D(K4)
6591 A6=F(K4)
6592 GOSUB 8000
6593 O5=A+O5
6595 IF O3=O5 THEN 6680
6600 O5=A
6601 A5=D(K3)
6602 A6=F(K3)
6603 GOSUB 8000
6604 O5=A+O5
6605 A1=D(K4)
6606 A2=F(K4)
6607 GOSUB 8000
6608 O5=A+O5
6610 IF O4>O5 THEN 6700
6620 L1=D(K4)
6621 L2=E(K4)
6622 L3=F(K4)
6623 L4=D(K2)
6624 L5=E(K2)
6625 L6=F(K2)
6630 GOSUB 9000
6635 IF L0>Z9 THEN 6700
6640 GO TO 6800
6650 L1=D(K1)
6651 L2=E(K1)
6652 L3=F(K1)
6653 L4=D(K3)
6654 L5=E(K3)
6655 L6=F(K3)
6656 L7=D(K4)
6657 L8=E(K4)
6658 L9=F(K4)
6660 GOSUB 9000
6665 IF L0<Z9 THEN 6800
6670 IF O9=0 THEN 6580
6675 GO TO 6700
6680 L4=D(K2)
6681 L5=E(K2)
6682 L6=F(K2)
6683 L7=D(K4)
6684 L8=E(K4)

```

6685 L9=F(K4)
6690 GOSUB 9000
6692 IF LO<Z9 THEN 6800
6700 NEXT I8
6710 NEXT I9
6715 IF G1=1 AND Y8>0.999 THEN 6850
6716 IF G1=1 THEN 6820
6717 G1=1
6718 MOVE @0:G0,H0
6720 GO TO 6820
6800 IF G1=0 THEN 6810
6805 DRAW @0:G0,H0
6806 G1=0
6810 IF Y8>0.999 THEN 6900
6820 G0=X9
6830 H0=Y9
6840 GO TO 6005
6850 DRAW @0:X9,Y9
6900 RETURN
7000 IF M1=M2 OR (D(J)=G5 AND J9=1) THEN 7100
7001 IF D(J)<>G5 THEN 7006
7002 IF F(J)=G7 THEN 7100
7003 X0=G5
7004 Y7=M2*(X0-D(K1))+F(K1)
7005 GO TO 7020
7006 IF J9=0 THEN 7010
7007 X0=D(K1)
7008 GO TO 7018
7010 X0=(F(K1)-G7+M1*G5-M2*D(K1))/(M1-M2)
7018 Y7=M1*(X0-D(J))+F(J)
7020 IF X0<=X7 OR X0=>X8 OR X0<=D8 OR X0=>D9 THEN 7100
7025 IF D(J)=G5 THEN 7042
7030 IF Y8=>(X0-G5)/(D(J)-G5) OR Y0<=(X0-G5)/(D(J)-G5) THEN 7100
7040 Y0=(X0-G5)/(D(J)-G5)
7041 GO TO 7050
7042 IF Y8=>(Y7-G7)/(F(J)-G7) OR Y0<=(Y7-G7)/(F(J)-G7) THEN 7100
7044 Y0=(Y7-G7)/(F(J)-G7)
7050 Y9=Y7
7055 X9=X0
7100 RETURN
8000 A=0.5*ABS(A1*(A4-A6)+A3*(A6-A2)+A5*(A2-A4))
8002 IF O9=0 OR O9=1 AND A=0 THEN 6700
8005 RETURN
9000 IF (L6-L3)*(L7-L1)=(L9-L3)*(L4-L1) THEN 6700
9002 L0=((L8-L2)*(L6-L3)-(L5-L2)*(L9-L3))*(Z7-L1)
9005 L0=L0+((L7-L1)*(L5-L2)-(L4-L1)*(L8-L2))*(Z8-L3)
9010 L0=L0/((L6-L3)*(L7-L1)-(L9-L3)*(L4-L1))+L2
9015 RETURN

```


Appendix C -- Bargraph Program Listing

```

1 GO TO 100
4 GO TO 1220
100 GO TO 170
110 REM THIS PROGRAM WAS WRITTEN BY LYSANDER NG OF ODPS,
120 REM NMFS-CHARLESTON FOR GRAPHICAL REPRESENTATION OF DATA
130 REM USING BAR CHARTS.
140 REM DATE WRITTEN : DECEMBER 1981
150 REM APRIL 1983 REVISION : STORY ADDED
160 REM MAY 1983 REVISION : 2 ENHANCEMENTS ADDED
170 INIT
180 DIM B$(200),G$(200),C(15),H(15),K(8),D(200),P$(200),N(8)
190 H$=CHR(8)
200 J$=CHR(10)
210 K$=CHR(11)
220 B$=" "
230 G$=" "
240 P$=" "
250 REM THIS FIRST PART OF THE PROGRAM REQUESTS USER FOR DATA ***
260 PRINT "INPUT Y RANGE (MINIMUM,MAXIMUM) : ";
270 INPUT V1,V2
280 PRINT "INPUT Y AXIS INCREMENT ";
290 INPUT V6
300 PRINT "INPUT CHARACTER SIZE FOR Y AXIS NUMERALS ";
310 INPUT V7,V8
320 PRINT " "
330 PRINT "INPUT NUMBER OF STORIES ";
340 INPUT S6
350 PRINT "INPUT LABEL FOR EACH STORY"
360 FOR I=1 TO S6
370 PRINT "STORY ";I;" ---- ";
380 INPUT M$
390 N(I)=LEN(M$)
400 P#=P$&M#
410 NEXT I
420 PRINT "INPUT CHARACTER SIZE FOR STORY LABELS ";
430 INPUT Y2,Y3
440 PRINT " "
450 PRINT "INPUT NUMBER OF BLOCKS AT EACH STORY ";
460 INPUT B1
470 PRINT "INPUT LABEL FOR EACH BLOCK"
480 FOR I=1 TO B1
490 PRINT "BLOCK ";I;" ---- ";
500 INPUT A$
510 B#=B$&A$
520 C(I)=LEN(A$)
530 NEXT I
540 PRINT "INPUT CHARACTER SIZE FOR BLOCK LABEL ";
550 INPUT B2,B3

```

```
560 PRINT " "
570 PRINT "INPUT NUMBER OF GROUPS IN EACH BLOCK ";
580 INPUT G1
590 PRINT "INPUT LABEL FOR EACH GROUP"
600 FOR I=1 TO G1
610 PRINT "GROUP ";I;" ---- ";
620 INPUT F$
630 G$=G$&F$
640 H(I+1)=LEN(F$)
650 NEXT I
660 PRINT "INPUT CHARACTER SIZE FOR GROUP LABEL ";
670 INPUT G2,G3
680 PRINT " "
690 PRINT "INPUT NUMBER OF TREATMENTS IN EACH GROUP ";
700 INPUT T1
710 PRINT "ENHANCEMENTS FOR EACH TREATMENT (INPUT CORRESPONDING #)"
720 PRINT "1 BLANK"
730 PRINT "2 CROSS-HATCHING"
740 PRINT "3 HORIZONTAL BARS"
750 PRINT "4 SOLID"
760 PRINT "5 CROSSES"
770 PRINT "6 STARS"
780 PRINT "7 STRIATED CROSS-HATCHING"
790 PRINT "8 STRIATED HORIZONTAL BARS"
800 PRINT " "
810 FOR I=1 TO T1
820 PRINT "TREATMENT ";I;" ---- ";
830 INPUT K(I)
840 NEXT I
850 PRINT " "
860 PRINT "INPUT Y LABEL"
870 INPUT X$
880 PRINT "INPUT CHARACTER SIZE FOR Y LABEL ";
890 INPUT X2,X3
900 PRINT " "
910 PRINT "INPUT GRAPH LABEL"
920 INPUT Z$
930 PRINT "INPUT CHARACTER SIZE FOR GRAPH LABEL ";
940 INPUT Z2,Z3
950 J=0
960 PRINT " "
970 PRINT " "
980 PRINT "INPUT YOUR DATA"
990 PRINT " "
1000 PRINT "DO YOU WISH TO GRAPH STANDARD DEVIATIONS? (Y,N) ";
1010 INPUT C$
1020 IF C$="N" THEN 1040
1030 DIM L(200)
1040 FOR S7=1 TO 56
1050 PRINT "STORY ";S7
```

```
1060 FOR B5=1 TO B1
1070 PRINT " BLOCK ";B5
1080 FOR G5=1 TO G1
1090 PRINT " GROUP ";G5
1100 FOR T5=1 TO T1
1110 PRINT " TREATMENT ";T5;" ---- ";
1120 J=J+1
1130 INPUT D(J)
1140 IF C$="N" THEN 1180
1150 PRINT " STANDARD DEVIATION ---- ";
1160 INPUT L(J)
1170 PRINT " "
1180 NEXT T5
1190 NEXT G5
1200 NEXT B5
1210 NEXT S7
1220 REM REQUEST FOR DATA FINISHED. PREPARE FOR PLOTTING. ***
1230 WINDOW 0,130,0,100
1240 VIEWPORT 0,130,0,100
1250 H(1)=1
1260 D$="N"
1270 Q=2
1280 S8=0
1290 I3=0
1300 PRINT "INPUT OUTPUT LOCATION : PLOTTER=1,SCREEN=32 ";
1310 INPUT O
1320 IF O=32 THEN 1360
1330 PRINT @0,7:
1340 PRINT "DO YOU WISH TO ROTATE CHARACTERS OF Y LABEL? (Y,N) ";
1350 INPUT D$
1360 PAGE
1370 S1=(V2-V1)*S6/20
1380 S2=S1/2
1390 S3=1000/(B1*(G1*(T1+1.5)+1.5))
1400 S3=INT(S3)
1410 S4=1/S3
1420 S3=S3-1
1430 REM START PRINTING LABELS
1440 REM PRINT GRAPH LABEL
1450 MOVE @0:70,93
1460 PRINT @0,17:Z2,Z3
1470 FOR I=1 TO LEN(Z$) STEP 2
1480 PRINT @0:H$;
1490 NEXT I
1500 PRINT @0:Z$;
1510 FOR S7=1 TO S6
1520 V=2
1530 VIEWPORT 20,120,99-S7*(80/S6),87-(S7-1)*(80/S6)
1540 WINDOW 0,B1*(G1*(T1+1.5)+1.5),V1,V2
1550 REM PRINT STORY LABELS
```

```

1560 PRINT @0,17:Y2,Y3
1570 MOVE @0:B1*(G1*(T1+1.5)+1.5)/2,V1
1580 M$=SEG(P$,Q,N(S7))
1590 Q=Q+N(S7)
1600 FOR I=1 TO N(S7) STEP 2
1610 PRINT @0:H$;
1620 NEXT I
1630 PRINT @0:J$;J$;
1640 PRINT @0:M$;
1650 REM PRINT NUMERALS ON Y AXIS
1660 FOR I=V1 TO V2 STEP V6
1670 MOVE @0:O,I-0.45*V8*(V2-V1)/(80/S6-12)
1680 I$=STR(I)
1690 AO=LEN(I$)
1700 PRINT @0,17:V7,V8
1710 FOR IO=1 TO AO
1720 S8=S8 MAX AO
1730 PRINT @0:H$;
1740 NEXT IO
1750 PRINT @0:I;
1760 NEXT I
1770 AXIS @0:O,V6,O,V1
1780 REM PRINT GROUP LABELS
1790 PRINT @0,17:G2,G3
1800 FOR I=1 TO B1
1810 W=1
1820 FOR J=1 TO G1+1
1830 MOVE @0:(J-1)*(1.5+T1)-T1/2+(I-1)*(G1*(T1+1.5)+1.5),V1
1840 FOR L1=1 TO H(J)
1850 RMOVE @0:-0.0045*G2*B1*(G1*(T1+1.5)+1.5),0
1860 NEXT L1
1870 PRINT @0:J$;
1880 F$=SEG(G$,W,H(J))
1890 W=W+H(J)
1900 PRINT @0:F$;
1910 NEXT J
1920 NEXT I
1930 REM PRINT BLOCK LABELS
1940 PRINT @0,17:B2,B3
1950 FOR I=1 TO B1
1960 MOVE @0:(G1*(T1+1.5)+1.5)*(I-0.5),V2
1970 FOR J=1 TO C(I) STEP 2
1980 PRINT @0:H$;
1990 NEXT J
2000 A$=SEG(B$,V,C(I))
2010 V=V+C(I)
2020 PRINT @0:A$;
2030 NEXT I
2040 REM LABEL PRINTING COMPLETED, EXCEPT FOR Y AXIS LABEL
2050 REM START TO PLOT BARS

```

```
2060 I4=-1
2070 FOR B5=1 TO B1
2080 FOR G5=1 TO G1
2090 I2=0
2100 I4=I4+1.5
2110 FOR T5=1 TO T1
2120 I3=I3+1
2130 I4=I4+1
2140 I2=I2+1
2150 MOVE @0:I4,V1
2160 RDRAW @0:0,D(I3)-V1
2170 RDRAW @0:1,0
2180 IF C$="N" THEN 2240
2190 RMOVE @0:-0.5,0
2200 RDRAW @0:0,L(I3)
2210 RMOVE @0:-0.25,0
2220 RDRAW @0:0.5,0
2230 RMOVE @0:0.25,-L(I3)
2240 IF I2<T1 THEN 2270
2250 RDRAW @0:0,V1-D(I3)
2260 GO TO 2300
2270 IF D(I3)<=D(I3+1) THEN 2300
2280 RDRAW @0:0,D(I3+1)-D(I3)
2290 REM SEND TO EXECUTE ENHANCEMENT ROUTINES
2300 GO TO K(I2) OF 2620
2310 GO TO K(I2)-1 OF 2370,2530,2480
2320 GO TO K(I2)-6 OF 2370,2530
2330 FOR I=1 TO 20
2340 MOVE @0:I4,D(I3)-I*S1+S1
2350 RDRAW @0:1,-S1
2360 NEXT I
2370 FOR I=1 TO 20
2380 J=0
2390 MOVE @0:I4+1,D(I3)-I*S1+S1-J
2400 RDRAW @0:-1,-S1
2410 IF K(I2)>7 THEN 2450
2420 IF J>S1/4 THEN 2450
2430 J=J+S1/20
2440 GO TO 2390
2450 NEXT I
2460 GO TO K(I2)-5 OF 2530
2470 GO TO 2620
2480 FOR I=1 TO S3
2490 MOVE @0:I4+S4*I,D(I3)
2500 DRAW @0:I4+S4*I,V1
2510 NEXT I
2520 GO TO 2620
2530 FOR I=1 TO 40
2540 J=0
2550 MOVE @0:I4,D(I3)-I*S2+J
```

```
2560 RDRAW @0:1,0
2570 IF K(I2)<>8 THEN 2610
2580 J=J+S1/20
2590 IF J>S1/5 THEN 2610
2600 GO TO 2550
2610 NEXT I
2620 NEXT T5
2630 NEXT G5
2640 I4=I4+1.5
2650 MOVE @0:I4+1,V1
2660 DRAW @0:I4+1,V2
2670 NEXT B5
2680 I4=I4+1.5
2690 NEXT S7
2700 WINDOW 0,130,0,100
2710 VIEWPORT 0,130,0,100
2720 MOVE @0:20,55
2730 REM PRINT Y AXIS LABEL
2740 PRINT @0,17:V7,V8
2750 FOR I=1 TO S8
2760 PRINT @0:H$;
2770 NEXT I
2780 PRINT @0,17:X2,X3
2790 IF D$="N" THEN 2870
2800 PRINT @0:H$;
2810 PRINT @0,25:90
2820 FOR I=1 TO LEN(X$) STEP 2
2830 PRINT @0:H$;
2840 NEXT I
2850 PRINT @0:X$;
2860 GO TO 2960
2870 FOR I=1 TO LEN(X$) STEP 2
2880 PRINT @0:K$;
2890 NEXT I
2900 PRINT @0:H$;
2910 FOR I=1 TO LEN(X$)
2920 M$=SEG(X$,I,1)
2930 PRINT @0:H$;J$;M$;
2940 NEXT I
2950 REM Y AXIS LABEL PRINTED
2960 MOVE 10,10
2970 PRINT "TO DRAW GRAPH AGAIN, PRESS USER DEFINABLE KEY #1."
2980 END
```

Appendix D -- Fractional Bargraph Program Listing

```
1 GO TO 160
4 GO TO 1000
120 REM THIS PROGRAM WAS WRITTEN BY LYSANDER NG OF ODPS,
130 REM NMFS-CHARLESTON FOR GRAPHICAL REPRESENTATION OF DATA
140 REM USING FRACTIONAL BAR CHARTS.
150 REM DATE WRITTEN : JUNE 1983
160 INIT
170 DIM B$(200),G$(200),C(15),H(15),K(8),D(200),P$(200),N(8)
180 H$=CHR(8)
190 J$=CHR(10)
200 K$=CHR(11)
210 L$=CHR(13)
220 B$=" "
230 G$=" "
240 P$=" "
250 REM THIS FIRST PART OF THE PROGRAM REQUESTS USER FOR DATA ***
260 PRINT "Y MINIMUM IS FIXED AT 0, INPUT Y MAXIMUM: ";
270 INPUT V2
280 PRINT "INPUT Y AXIS INCREMENT ";
290 INPUT V6
300 PRINT "INPUT CHARACTER SIZE FOR Y AXIS NUMERALS ";
310 INPUT V7,V8
320 PRINT L$;"INPUT NUMBER OF STORIES ";
330 INPUT S6
340 PRINT "INPUT LABEL FOR EACH STORY"
350 FOR I=1 TO S6
360 PRINT "STORY ";I;" ---- ";
370 INPUT M$
380 N(I)=LEN(M$)
390 P$=P$&M$
400 NEXT I
410 PRINT "INPUT CHARACTER SIZE FOR STORY LABELS ";
420 INPUT Y2,Y3
430 PRINT L$;"INPUT NUMBER OF BLOCKS AT EACH STORY ";
440 INPUT B1
450 PRINT "INPUT LABEL FOR EACH BLOCK"
460 FOR I=1 TO B1
470 PRINT "BLOCK ";I;" ---- ";
480 INPUT A$
490 B$=B$&A$
500 C(I)=LEN(A$)
510 NEXT I
520 PRINT "INPUT CHARACTER SIZE FOR BLOCK LABEL ";
530 INPUT B2,B3
540 PRINT L$;"INPUT NUMBER OF GROUPS IN EACH BLOCK ";
550 INPUT G1
560 PRINT "INPUT LABEL FOR EACH GROUP"
570 FOR I=1 TO G1
```

```
580 PRINT "GROUP ";I;" ---- ";
590 INPUT F$
600 G$=G$&F$
610 H(I+1)=LEN(F$)
620 NEXT I
630 PRINT "INPUT CHARACTER SIZE FOR GROUP LABEL ";
640 INPUT G2,G3
650 PRINT L$;"INPUT NUMBER OF SEGMENTS IN EACH GROUP ";
660 INPUT T1
670 PRINT "ENHANCEMENT FOR EACH SEGMENT (INPUT CORRESPONDING #)"
680 PRINT "1 BLANK"
690 PRINT "2 STRIPES"
700 PRINT "3 VERTICLE LINES"
710 PRINT "4 SOLID"
720 FOR I=1 TO T1
730 PRINT "SEGMENT ";I;" ---- ";
740 INPUT K(I)
750 NEXT I
760 PRINT L$;"INPUT Y LABEL"
770 INPUT X$
780 PRINT "INPUT CHARACTER SIZE FOR Y LABEL ";
790 INPUT X2,X3
800 PRINT L$;"INPUT GRAPH LABEL"
810 INPUT Z$
820 PRINT "INPUT CHARACTER SIZE FOR GRAPH LABEL ";
830 INPUT Z2,Z3
840 J=0
850 PRINT L$,L$;"INPUT YOUR DATA",L$
860 FOR S7=1 TO S6
870 PRINT "STORY ";S7
880 FOR B5=1 TO B1
890 PRINT " BLOCK ";B5
900 FOR G5=1 TO G1
910 PRINT " GROUP ";G5
920 FOR T5=1 TO T1
930 PRINT " SEGMENT ";T5;" ---- ";
940 J=J+1
950 INPUT D(J)
960 NEXT T5
970 NEXT G5
980 NEXT B5
990 NEXT S7
1000 REM REQUEST FOR DATA FINISHED. PREPARE FOR PLOTTING. ***
1010 WINDOW 0,130,0,100
1020 VIEWPORT 0,130,0,100
1030 H(1)=1
1040 D$="N"
1050 Q=2
1060 S8=0
1070 I3=0
```

```
1080 PRINT " "
1090 PRINT "INPUT OUTPUT LOCATION : PLOTTER=1, SCREEN=32 ";
1100 INPUT O
1110 IF O=32 THEN 1150
1120 PRINT @0,7:
1130 PRINT "DO YOU WISH TO ROTATE CHARACTERS OF Y LABEL? (Y,N) ";
1140 INPUT D$ 
1150 PAGE
1160 S5=INT(30/((2*G1+1)*B1)+1)
1170 S1=1/(2*S5+1)
1180 S3=600/(B1*(G1*2+1))
1190 S3=INT(S3)
1200 S4=1/S3
1210 S3=S3-1
1220 REM START PRINTING LABELS
1230 REM PRINT GRAPH LABEL
1240 MOVE @0:70,93
1250 PRINT @0,17:Z2,Z3
1260 FOR I=1 TO LEN(Z$) STEP 2
1270 PRINT @0:H$;
1280 NEXT I
1290 PRINT @0:Z$;
1300 FOR S7=1 TO S6
1310 V=2
1320 V4=99-S7*(80/S6)
1330 V3=87-(S7-1)*(80/S6)
1340 VIEWPORT 20,120,V4,V3
1350 WINDOW 0,B1*(G1*2+1),0,V2
1360 REM PRINT STORY LABELS
1370 PRINT @0,17:Y2,Y3
1380 MOVE @0:B1*(G1*2+1)/2,0
1390 M$=SEG(P$,Q,N(S7))
1400 Q=Q+N(S7)
1410 FOR I=1 TO N(S7) STEP 2
1420 PRINT @0:H$;
1430 NEXT I
1440 PRINT @0:J$,J$;
1450 PRINT @0:M$;
1460 REM PRINT NUMERALS ON Y AXIS
1470 FOR I=0 TO V2 STEP V6
1480 MOVE @0:0,I-0.45*V8*V2/(80/S6-12)
1490 I$=STR(I)
1500 AO=LEN(I$)
1510 PRINT @0,17:V7,V8
1520 FOR IO=1 TO AO
1530 S8=S8 MAX AO
1540 PRINT @0:H$;
1550 NEXT IO
1560 PRINT @0:I;
1570 NEXT I
```

```
1580 AXIS @0:0,V6,0,0
1590 REM PRINT GROUP LABELS
1600 PRINT @0,17:B2,B3
1610 FOR I=1 TO B1
1620 W=1
1630 FOR J=1 TO G1+1
1640 MOVE @0:(J-1)*2-0,5+(I-1)*(G1*2+1),0
1650 FOR L1=1 TO H(J)
1660 RMOVE @0:-0.0045*B2*B1*(G1*2+1),0
1670 NEXT L1
1680 PRINT @0:J$;
1690 F$=SEG(G$,W,H(J))
1700 W=W+H(J)
1710 PRINT @0:F$;
1720 NEXT J
1730 NEXT I
1740 REM PRINT BLOCK LABELS
1750 PRINT @0,17:B2,B3
1760 FOR I=1 TO B1
1770 MOVE @0:(G1*2+1)*(I-0.5),V2+0.005*V2*S6
1780 FOR J=1 TO C(I) STEP 2
1790 PRINT @0:H$;
1800 NEXT J
1810 A$=SEG(B$,V,C(I))
1820 V=V+C(I)
1830 PRINT @0:A$;
1840 NEXT I
1850 REM LABEL PRINTING COMPLETED, EXCEPT FOR Y AXIS LABEL
1860 REM START TO PLOT BARS
1870 I4=0
1880 FOR B5=1 TO B1
1890 FOR G5=1 TO G1
1900 I2=0
1910 I4=I4+1
1920 D1=0
1930 FOR T5=1 TO T1
1940 I3=I3+1
1950 I2=I2+1
1960 MOVE @0:I4,D1
1970 RDRAW @0:0,D(I3)
1980 RDRAW @0:1,0
1990 REM SEND TO EXECUTE ENHANCEMENT ROUTINES
2000 IF K(I2)=1 OR K(I2)=4 THEN 2110
2005 K1=1
2010 FOR I=1 TO 2*S5+1
2015 K1=NOT(K1)
2020 MOVE @0:I4+I*S1,D1
2030 RDRAW @0:0,D(I3)
2035 K2=INT(S3/(2*S5+1)+1)
2040 IF K(I2)=3 OR K1=1 THEN 2090
```

```
2050 FOR J=1 TO K2
2060 RMOVE @0:-S1/(K2+1),-D(I3)
2070 RDRAW @0:0,D(I3)
2080 NEXT J
2090 NEXT I
2100 GO TO 2170
2110 RDRAW @0:0,-D(I3)
2120 IF K(I2)=1 THEN 2170
2130 FOR I=1 TO S3
2140 MOVE @0:I4+S4*I,D(I3)+D1
2150 DRAW @0:I4+S4*I,D1
2160 NEXT I
2170 D1=D1+D(I3)
2180 NEXT T5
2190 I4=I4+1
2200 NEXT G5
2210 I4=I4+1
2220 NEXT B5
2230 MOVE @0:I4,0
2240 DRAW @0:I4,V2
2250 NEXT S7
2260 WINDOW 0,130,0,100
2270 VIEWPORT 0,130,0,100
2280 MOVE @0:20,55
2290 REM PRINT Y AXIS LABEL
2300 PRINT @0,17:V7,V8
2310 FOR I=1 TO S8
2320 PRINT @0:H$;
2330 NEXT I
2340 PRINT @0,17:X2,X3
2350 IF D$="N" THEN 2430
2360 PRINT @0:H$;
2370 PRINT @0,25:90
2380 FOR I=1 TO LEN(X$) STEP 2
2390 PRINT @0:H$;
2400 NEXT I
2410 PRINT @0:X$;
2420 GO TO 2520
2430 FOR I=1 TO LEN(X$) STEP 2
2440 PRINT @0:K$;
2450 NEXT I
2460 PRINT @0:H$;
2470 FOR I=1 TO LEN(X$)
2480 M$=SEG(X$,I,1)
2490 PRINT @0:H$,J$,M$;
2500 NEXT I
2510 REM Y AXIS LABEL PRINTED
2520 MOVE 10,10
2530 PRINT "TO DRAW GRAPH AGAIN, PRESS USER DEFINABLE KEY #1."
2540 END
```


Appendix E -- Legend Maker Program Listing

```

100 REM AUTHOR:LYSANDER NG, AFFILIATION:NMFS-CHARLESTON DATE:5/84
110 REM THIS PROGRAM IS WRITTEN TO COMPLEMENT THE BARGRAPH AND THE
120 REM FRACTIONAL BARGRAPH PLOTTING PROGRAMS.
130 REM IT PRODUCES LEGENDS FOR THEM.
132 DIM E(10),L(10),C$(800)
134 C$=""
140 PRINT "THE FOLLOWING ARE THE ENHANCEMENTS AVAILABLE:"
150 PRINT ""
160 PRINT "1. BLANK"
170 PRINT "2. CROSS-HATCHING"
180 PRINT "3. HORIZONTAL BARS"
190 PRINT "4. SOLID"
200 PRINT "5. CROSSES"
210 PRINT "6. STARS"
212 PRINT "7. STRIATED CROSS-HATCHING"
215 PRINT "8. STRIATED HORIZONTAL BARS"
220 PRINT "9. VERTICAL LINES"
230 PRINT "10. STRIPES"
240 PRINT ""
250 PRINT "ENTER THE TOTAL NUMBER OF ENHANCEMENTS IN YOUR LEGEND. ";
260 INPUT N
262 PRINT ""
264 PRINT "ENTER THE CORRESPONDING NUMBER OF ENHANCEMENT FOLLOW BY ";
266 PRINT "CAPTION."
270 FOR I=1 TO N
275 PRINT ""
280 PRINT "ENHANCEMENT ";I;" ... ";
285 INPUT E(I)
290 PRINT "CAPTION ";I;" : ";
300 INPUT D$
310 L(I)=LEN(D$)
330 C$=C$&D$
340 NEXT I
350 PRINT ""
352 PRINT "ENTER WIDTH OF CHARACTERS. FOR CHARACTER OF WIDTH ONE,"
354 PRINT "150 CAN BE FITTED IN ONE LINE ON PLOTTER. CHARACTER SIZE"
356 PRINT "ON SCREEN IS FIXED. ";
358 INPUT W
359 PRINT ""
360 PRINT "ENTER OUTPUT DEVICE. 1=PLOTTER; 32=SCREEN. ";
370 INPUT O
380 I1=1
520 PAGE
530 H1=188/(N*2.82)
540 H2=100/N
550 W1=W*1.55/1.79
560 PRINT @0,17:W,H2
570 FOR I=1 TO N

```

```
580 MOVE @0:0,100*(N-I)/N
582 RDRAW @0:W1,0
584 RDRAW @0:0,H1
586 RDRAW @0:-W1,0
588 RDRAW @0:0,-H1
590 GO TO E(I) OF 1000,2000,2750,4000,2000,2000,2000,2750,3000,4000
1000 D$=SEG(C$,I1,L(I))
1010 RMOVE @0:W*1.1,0
1020 PRINT @0:D$
1030 I1=I1+L(I)
1040 NEXT I
1050 PRINT "PLOT AGAIN TYPE `RUN 360`."
1060 END
2000 RDRAW @0:W1,H1
2010 RMOVE @0:-W1,-H1
2015 IF E(I)=7 THEN 2200
2020 IF E(I)>2 THEN 2500
2030 GO TO 1000
2200 FOR I3=1 TO 9
2210 RMOVE @0:W1*(1.1-I3/10),H1*(1-I3/10)
2220 RDRAW @0:-W1*(1-I3/10),-H1*(1-I3/10)
2230 NEXT I3
2235 RMOVE -W1*0.9,0
2240 GO TO 1000
2500 RMOVE @0:W1,0
2510 RDRAW @0:-W1,H1
2520 RMOVE @0:0,-H1
2530 IF E(I)=6 THEN 2750
2540 GO TO 1000
2750 RMOVE @0:0,H1*0.5
2760 RDRAW @0:W1,0
2770 RMOVE @0:-W1,-H1*0.5
2775 IF E(I)=8 THEN 2800
2780 GO TO 1000
2800 FOR I3=1 TO 5
2810 RMOVE @0:0,H1*0.05
2820 RDRAW @0:W1,0
2830 RMOVE @0:0,H1*0.05
2840 RDRAW @0:-W1,0
2850 NEXT I3
2855 RMOVE 0,-H1*0.5
2860 GO TO 1000
3000 RMOVE @0:W1*0.5,0
3010 RDRAW @0:0,H1
3020 RMOVE @0:-W1*0.5,-H1
3030 GO TO 1000
4000 I2=1
4010 FOR I3=1 TO 5
4020 RMOVE @0:W1*0.05,0
4030 RDRAW @0:0,H1
```

```
4040 RMOVE @0:W1*0.05,0
4050 RDRAW @0:0,-H1
4060 NEXT I3
4070 RMOVE @0:-W1*I2*0.5,0
4080 IF I2=2 THEN 1000
4090 I2=2
4100 IF E(I)=10 THEN 1000
4110 RMOVE @0:W1*0.5,0
4120 GO TO 4010
```


Appendix F -- X-Y Graph Program Listing

```

100 REM THIS IS AN X-Y PLOTTING ROUTINE OF THE PLOT-50 PACKAGE
110 REM MODIFIED BY LYSANDER NG OF ODPS NMFS-CHARLESTON
115 REM LAST REVISION : 6/21/83
120 INIT
130 PRINT @32,17:4
140 DIM T(23),X(300),Y(300),U$(20),R$(1)
150 DIM E2(300),T3(15),M3(15),T2(15)
160 D=32
170 G$=CHR(7)
180 H$=CHR(8)
190 I$=CHR(9)
200 A$=CHR(10)
210 X$=" "
220 K$=CHR(11)
230 L$=CHR(12)
240 O$=CHR(13)
250 Y$=" "
251 FOR I=1 TO 300
252 E2(I)=0
253 NEXT I
260 X=0
270 M3(1)=1
280 N1=1
290 T=0
300 R=0
310 T(2)=2
320 T(3)=1
330 T(4)=30
340 T(5)=85
350 T(8)=1
360 T(9)=1
370 T(14)=15
380 T(15)=85
390 T(18)=1
400 T(19)=1
410 FOR N2=1 TO 15
420 T2(N2)=2
430 T3(N2)=1
440 M3(N2)=1
450 NEXT N2
460 PRINT L$,I$;"X vs Y DATA PLOT";O$
470 PRINT " 1 ENTER DATA";O$;" 2 DISPLAY DATA"
480 PRINT " 3 LIST DATA";O$;" 4 LIST PARAMETERS"
490 PRINT " 5 SET AUTOSCALE";O$;" 6 SET X SCREEN POSITION"
500 PRINT " 7 SET Y SCREEN POSITION";O$;" 8 SET X DATA RANGE"
510 PRINT " 9 SET Y DATA RANGE";O$;"10 SELECT PLOT MODE"
520 PRINT "11 SELECT SYMBOL";O$;"12 INSERT DATA";O$;"13 DELETE DATA"
530 PRINT "14 CHANGE DATA";O$;"15 STORE DATA";O$;"16 STOP"

```

```

540 PRINT "17 CHANGE Y VALUES ONLY"
550 PRINT O$;O$;"ENTER MENU ITEM NUMBER OR M FOR NEW MENU ";G$;
560 GOSUB 610
570 U1=VAL(U$)
580 IF U1<=17 THEN 660
590 PRINT "ERROR";G$
600 GO TO 550
610 INPUT U$
620 IF U$="M" THEN 460
630 IF U$="N" THEN 1480
640 U$=U$&",0"
650 RETURN
660 IF U1<1 THEN 590
670 IF U1>11 THEN 710
680 IF U1>5 THEN 700
690 GO TO U1 OF 1340,2900,2800,2470,870
700 GO TO U1-5 OF 960,940,950,930,1120,1270
710 GO TO U1-11 OF 1860,1860,1860,750,730,1340
720 GO TO 460
730 FIND 1
740 OLD
750 PRINT "INSERT DATA TAPE AND ENTER FILE NUMBER";
760 GOSUB 610
770 U1=VAL(U$)
780 FIND U1
790 WRITE T(1),N1
792 FOR U1=1 TO N1
794 WRITE M3(U1)
795 NEXT U1
800 FOR U1=1 TO T(1)
810 WRITE X(U1),Y(U1),E2(U1)
820 NEXT U1
830 CLOSE
840 PRINT "DATA STORED";G$;O$;"INSERT SYSTEM TAPE AND PRESS RETURN";
850 GOSUB 610
860 GO TO 550
870 T(6)=0
880 T(7)=0
890 T(16)=0
900 T(17)=0
910 PRINT "AUTOSCALED"
920 GO TO 550
930 R=2
940 R=R+8
950 R=R+2
960 R=R+4
970 U$=CHR(88+(R>11))
980 PRINT "PLEASE ENTER ";U$;
990 U$="SCREEN"
1000 IF R=4 OR R=14 THEN 1020

```

```

1010 U$="DATA"
1020 PRINT " ";U$
1030 PRINT " ","MINIMUM ";
1040 GOSUB 610
1050 T(R)=VAL(U$)
1060 PRINT " ","MAXIMUM ";
1070 GOSUB 610
1080 T(R+1)=VAL(U$)
1090 T(R+1)=T(R+1)-T(R)
1100 R=0
1110 GO TO 550
1120 PRINT 0$;"1 LINE PLOT MODE";0$;"2 POINT PLOT MODE"
1130 PRINT "3 DASH PLOT MODE";0$;"4 DOT PLOT MODE";0$
1140 FOR N2=1 TO N1
1150 PRINT 0$;"PLOT MODE FOR CURVE ";N2
1160 PRINT "SELECT NUMBER AND PRESS RETURN ";
1170 GOSUB 610
1180 T(U1-8)=VAL(U$)
1190 IF U1<>11 THEN 1220
1200 T3(N2)=T(3)
1210 GO TO 1320
1220 IF U1<>10 THEN 1250
1230 T2(N2)=T(2)
1240 NEXT N2
1250 GO TO 550
1260 FOR N2=1 TO N1
1270 PRINT 0$;"1 POINT";0$;"2 TRIANGLE";0$;"3 PLUS SIGN"
1280 PRINT "4 SQUARE";0$;"5 DIAMOND";0$;"6 CIRCLE"
1290 FOR N2=1 TO N1
1300 PRINT 0$;"SYMBOL FOR CURVE ";N2
1310 GO TO 1160
1320 NEXT N2
1330 GO TO 550
1340 T(6)=0
1350 T(7)=0
1360 T(16)=0
1370 T(17)=0
1380 IF T(1)<=0 THEN 1500
1390 PRINT 0$;" DO YOU WISH TO ADD TO PREVIOUS DATA? ";
1400 GOSUB 1670
1410 GO TO U0 OF 1450,1420
1420 PRINT 0$;"NEW CURVE? ";
1430 GOSUB 1670
1440 GO TO U0 OF 1500,1480
1450 LET T(1)=0
1460 N1=1
1462 FOR I=1 TO 15
1464 M3(I)=1
1466 NEXT I
1470 GO TO 1500

```

```

1480 N1=N1+1
1490 M3(N1)=T(1)+1
1500 PRINT "KEYBOARD?"; 
1510 GOSUB 1670
1520 GO TO U0 OF 1720,1530
1530 PRI L$;"ENTER M TO DISCONTINUE DATA ENTRY; N FOR ANOTHER CURVE";0$
1540 U9=T(1)+1
1550 IF U1=17 THEN 1620
1560 IF U9<=300 THEN 1590
1570 PRINT "MAXIMUM NUMBER OF VALUES IS 300";G$
1580 GO TO 550
1590 PRINT "X(";U9;")   ";
1600 GOSUB 610
1610 X(U9)=VAL(U$)
1620 PRINT "Y(";U9;")   ";
1630 GOSUB 610
1640 Y(U9)=VAL(U$)
1650 T(1)=U9
1660 GO TO 1540
1670 PRINT "(Y or N) ";
1680 INPUT R$
1690 U0=POS("NY",R$,1)
1700 IF U0=0 THEN 1680
1710 RETURN
1720 PRINT "TAPE PROCESSING";
1730 GOSUB 1670
1740 GO TO U0 OF 1470,1750
1750 PRINT "INSERT DATA TAPE AND ENTER TAPE FILE # ";
1760 GOSUB 610
1770 U1=VAL(U$)
1780 FIND U1
1790 READ @33:M5,M6,M7
1791 DELETE M7
1792 IF M6=1 THEN 1800
1793 FOR U1=N1+1 TO N1+M6-1
1794 READ @33:M3(U1)
1795 M3(U1)=M3(U1)+T(1)
1796 NEXT U1
1800 FOR U1=T(1)+1 TO T(1)+M5
1810 READ @33:X(U1),Y(U1),E2(U1)
1820 NEXT U1
1822 T(1)=T(1)+M5
1824 N1=N1+M6-1
1830 PRINT "INSERT SYSTEM TAPE AND PRESS RETURN";G$;
1840 GOSUB 610
1850 GO TO 550
1860 U$=SEG("PRECEDEDDELETED CHANGED ",(U1-12)*8+1,8)
1870 T(6)=0
1880 T(7)=0
1890 T(16)=0

```

```

1900 T(17)=0
1910 PRINT 0$;0$;0$;0$;"ENTER ITEM NUMBER TO BE ";U$;G$
1920 PRINT "ENTER 0 ONLY TO EXIT."
1930 GOSUB 610
1940 U2=VAL(U$)
1950 IF U2=>1 AND U2<=T(1) THEN 1980
1960 PRINT "OUT OF RANGE";G$;G$
1970 GO TO 550
1980 GO TO U1-11 OF 1990,2290,2420
1990 IF T(1)=>300 THEN 1960
2000 PRINT "ENTER NEW X VALUE ";
2010 GOSUB 610
2020 U4=VAL(U$)
2030 PRINT "ENTER NEW Y VALUE ";
2040 GOSUB 610
2050 U3=VAL(U$)
2060 FOR U1=U2 TO T(1)
2070 U5=X(U1)
2080 X(U1)=U4
2090 U4=U5
2100 U5=Y(U1)
2110 Y(U1)=U3
2120 U3=U5
2130 NEXT U1
2140 FOR N2=1 TO N1
2150 IF M3(N2)<U2 THEN 2230
2160 IF M3(N2)>U2 THEN 2210
2170 PRINT "DO YOU WANT TO INSERT INTO CURVE ",N2-1," OR ",N2," ? "
2180 INPUT M5
2190 IF M5=N2 THEN 2220
2200 M3(N2)=M3(N2)+1
2210 IF N2=15 THEN 2230
2220 M3(N2+1)=M3(N2+1)+1
2230 NEXT N2
2240 T(1)=T(1)+1
2250 X(T(1))=U4
2260 Y(T(1))=U3
2270 U1=12
2280 GO TO 1860
2290 PRINT "VALUE DELETED: X ",X(U2)," Y ",Y(U2)
2300 IF U2=T(1) THEN 2390
2310 FOR U1=U2 TO T(1)-1
2320 X(U1)=X(U1+1)
2330 Y(U1)=Y(U1+1)
2340 NEXT U1
2350 FOR N2=1 TO N1
2360 IF M3(N2)<=U2 THEN 2380
2370 M3(N2)=M3(N2)-1
2380 NEXT N2
2390 T(1)=T(1)-1

```

```

2400 U1=13
2410 GO TO 1860
2420 PRINT "CURRENT VALUE OF ITEM ",U2," IS X ",X(U2)," Y ",Y(U2)
2430 PRINT "ENTER NEW X AND Y VALUES";
2440 INPUT X(U2),Y(U2)
2450 U1=14
2460 GO TO 1860
2470 PRINT L$;" PARAMETER LIST"
2480 FOR N2=1 TO N1
2490 PRINT 0$;"FOR CURVE ",N2;0$;"NUMBER OF POINTS = ";
2500 IF N2<N1 THEN 2530
2510 PRINT T(1)-M3(N2)+1
2520 GO TO 2540
2530 PRINT M3(N2+1)-M3(N2)
2540 PRINT "LINE CODE = ";T2(N2)
2550 PRINT "SYMBOL CODE = ";T3(N2)
2560 NEXT N2
2570 U9=4
2580 U8=14
2590 U2=0
2600 PRINT
2610 FOR U5=1 TO 20
2620 NEXT U5
2630 PRINT I$;"-X-           -Y-"
2640 U$="SCREEN MINIMUM"
2650 GOSUB 2730
2660 U$="SCREEN RANGE"
2670 GOSUB 2730
2680 U$="DATA MINIMUM"
2690 GOSUB 2730
2700 U$="DATA RANGE"
2710 GOSUB 2730
2720 GO TO 550
2730 U5=20-LEN(U$)
2740 PRINT U$;
2750 FOR U1=1 TO U5
2760 NEXT U1
2770 PRINT I$,T(U9+U2),T(U8+U2)
2780 U2=U2+1
2790 RETURN
2800 PRINT L$;I$;"DATA LIST"
2810 PRINT "NUMBER          X          Y"
2820 U9=T(1)
2830 IF U9=0 THEN 2880
2840 FOR U1=1 TO U9
2850 PRINT U1,X(U1),Y(U1)
2860 NEXT U1
2870 GO TO 550
2880 PRINT "NO DATA POINTS"
2890 GO TO 550

```

```
2900 PRINT "LABEL X (Y OR N)?"
2910 INPUT S$
2920 IF S$="N" THEN 2950
2930 PRINT "ENTER X LABEL"
2940 INPUT X$
2950 PRINT "LABEL Y (Y OR N)?"
2960 INPUT S$
2970 IF S$="N" THEN 3000
2980 PRINT "ENTER Y LABEL"
2990 INPUT Y$
3000 PRINT "WOULD YOU LIKE A LINEAR LEAST SQUARE FIT (Y or N)?";
3010 INPUT S$
3020 IF S$="N" THEN 3110
3030 PRINT "OUTPUT OF LINEAR REGRESSION RESULTS (1-PLOTTER, 32-CRT)?"
3040 INPUT D
3050 X8=0
3060 Y8=0
3070 Y9=0
3080 X7=0
3090 Y7=0
3100 GO TO 3130
3110 T(14)=10
3120 T(15)=80
3130 PRINT "WOULD YOU LIKE TO MARK OFF THE STANDARD DEVIATIONS?";
3140 INPUT C$
3150 IF C$="N" THEN 3230
3160 PRINT "NEW STANDARD DEVIATIONS?"
3170 INPUT D$
3180 IF D$="N" THEN 3230
3190 FOR J=1 TO T(1)
3200 PRINT "STANDARD DEVIATION #";J; "...";
3210 INPUT E2(J)
3220 NEXT J
3230 IF T(17)>0 THEN 3250
3240 GOSUB 5110
3250 U9=4
3260 IF T(7)>0 THEN 3280
3270 GOSUB 5200
3280 GOSUB 5290
3290 U9=14
3300 GOSUB 5290
3310 U8=4
3320 PRINT @32;"OUTPUT DEVICE OF GRAPH, AXIS (CRT=32, PLOTTER=1)?"
3330 INPUT Z,V
3340 W$="N"
3350 N1=1
3360 IF V=32 THEN 3390
3370 PRINT "PRINT Y LABEL WITH ROTATED CHARACTERS (Y OR N)?"
3380 INPUT W$
3390 PAGE
```

```

3400 VIEWPORT T(4),T(4)+T(5),T(14),T(14)+T(15)
3410 WINDOW T(6),T(6)+T(7),T(16),T(16)+T(17)
3420 AXIS @V:T(8),T(18)
3430 MOVE @V:(T(6)+T(6)+T(7))/2,T(16)
3440 PRINT @V:A$;A$;A$;
3450 FOR I=1 TO LEN(X$) STEP 2
3460 PRINT @V:H$;
3470 NEXT I
3480 PRINT @V:X$;
3490 MOVE @V:T(6),(T(16)+T(16)+T(17))/2
3500 IF V=32 THEN 3600
3510 IF W$="N" THEN 3600
3520 PRINT @V:H$,H$,H$,H$,H$,H$;
3530 PRINT @1,25:90
3540 FOR I=1 TO LEN(Y$) STEP 2
3550 PRINT @V:H$;
3560 NEXT I
3570 PRINT @V:Y$;
3580 PRINT @1,25:0
3590 GO TO 3680
3600 FOR I=1 TO LEN(Y$) STEP 2
3610 PRINT @V:K$;
3620 NEXT I
3630 PRINT @V:H$,H$,H$,H$,H$,H$,H$,A$;
3640 FOR I=1 TO LEN(Y$)
3650 J$=SEG(Y$,I,1)
3660 PRINT @V:J$;H$,A$;
3670 NEXT I
3680 U1=1
3690 R2=T(18)/(T(17)/T(15))
3700 R3=T(14)
3710 U$=STR(T(16)+(U1-1)*T(18))
3720 PRINT @V,21:0 MAX T(4)-(LEN(U$)+1)*1.78,R3-0.89
3730 PRINT @V:U$;
3740 IF U1>T(19) THEN 3780
3750 R3=R3+R2
3760 U1=U1+1
3770 GO TO 3710
3780 R2=T(8)/(T(7)/T(5))
3790 R3=T(4)
3800 U1=1
3810 U$=STR(T(6)+(U1-1)*T(8))
3820 PRINT @V,21:0 MAX R3-(LEN(U$)+1)*1.78*0.5,T(14)-4
3830 PRINT @V:U$;
3840 IF U1>T(9) THEN 3880
3850 R3=R3+R2
3860 U1=U1+1
3870 GO TO 3810
3880 FOR U1=1 TO T(1)
3890 IF S$="N" THEN 3980

```

```

3900 X8=X8+X(U1)
3910 Y8=Y8+Y(U1)
3920 X9=X(U1)*Y(U1)
3930 Y9=Y9+X9
3940 X6=X(U1)*X(U1)
3950 Y6=Y(U1)*Y(U1)
3960 X7=X7+X6
3970 Y7=Y7+Y6
3980 IF U1=1 THEN 4020
3990 IF M3(N1+1)=U1 THEN 4010
4000 D0=(T(17)-T(16))/100
4005 GO TO T2(N1) OF 4040, 4020, 4070, 4070
4010 N1=N1+1
4020 MOVE @Z:X(U1),Y(U1)
4030 GO TO 4360
4040 DRAW @Z:X(U1),Y(U1)
4050 GO TO 4360
4070 D1=((X(U1)-X(U1-1))^2+(Y(U1)-Y(U1-1))^2)^0.5
4074 D7=((X(U1)-X(U1-1))*(T(17)-T(16))/(T(7)-T(6)))^2
4075 D7=(D7+(Y(U1)-Y(U1-1))^2)^0.5
4077 D0=D0*D1/D7
4080 IF X(U1)<>X(U1-1) THEN 4140
4090 IF Y(U1)>Y(U1-1) THEN 4120
4100 D2=1.5*PI
4110 GO TO 4200
4120 D2=0.5*PI
4130 GO TO 4200
4140 IF Y(U1)<>Y(U1-1) OR X(U1)>X(U1-1) THEN 4170
4150 D2=PI
4160 GO TO 4200
4170 D2=ATN((Y(U1)-Y(U1-1))/(X(U1)-X(U1-1)))
4180 IF X(U1)-X(U1-1)>0 THEN 4200
4190 D2=D2+PI
4200 D3=D0*COS(D2)
4210 D4=D0*SIN(D2)
4220 D5=1
4230 D6=0
4240 D6=D6+D0
4250 IF D6>D1 AND D5=1 AND T2(N1)=3 THEN 4040
4260 IF D6>D1 THEN 4020
4265 D5=NOT(D5)
4270 GO TO T2(N1)-2 OF 4290, 4320
4290 GO TO D5+1 OF 4300, 4320
4300 RDRAW @Z:D3,D4
4310 GO TO 4240
4320 RMOVE @Z:D3,D4
4330 GO TO T2(N1)-2 OF 4240, 4340
4340 RDRAW @Z:0,0
4350 GO TO 4240
4360 GOSUB 4660

```

```

4370 NEXT U1
4380 IF C$="n" THEN 4490
4390 S2=T(4+3)/T(4+1)*0.4
4400 FOR J=1 TO T(1)
4410 MOVE @Z:X(J),Y(J)
4420 RMOVE @Z:-S2,-E2(J)
4430 RDRAW @Z:2*S2,0
4440 RMOVE @Z:-S2,0
4450 RDRAW @Z:0,2*E2(J)
4460 RMOVE @Z:-S2,0
4470 RDRAW @Z:2*S2,0
4480 NEXT J
4490 IF S$="N" THEN 4640
4500 R0=Y9-X8*Y8/T(1)
4510 R0=R0/SQR((X7-X8*X8/T(1))*(Y7-Y8*Y8/T(1)))
4520 B=(T(1)*Y9-X8*Y8)/(T(1)*X7-X8*X8)
4530 A=(Y8-B*X8)/T(1)
4540 Y9=(T(6)+T(7))*B+A
4550 Y8=A+B*T(6)
4560 PRINT @Z,21:
4570 MOVE @Z:T(6),Y8
4580 DRAW @Z:T(6)+T(7),Y9
4590 PRINT @D,21:20,0
4600 PRINT @D,17:1.5,2.5
4610 PRINT @D:"REGRESSION:Y=";A;"+";B;"X, CORR:";
4620 PRINT @D: USING "2D.5D":R0;
4630 PRINT @D,7:
4640 HOME
4650 GO TO 560
4660 S2=T(14+3)/T(14+1)*0.75
4670 S1=T(4+3)/T(4+1)*0.75
4680 GO TO T3(N1) OF 5080,4700,4760,4830,4900,4970
4690 GO TO 5080
4700 RMOVE @Z:0,S2
4710 RDRAW @Z:S1,-2*S2
4720 RDRAW @Z:-2*S1,0
4730 RDRAW @Z:S1,2*S2
4740 RMOVE @Z:0,-S2
4750 RETURN
4760 RDRAW @Z:0,S2
4770 RDRAW @Z:0,-2*S2
4780 RDRAW @Z:0,S2
4790 RDRAW @Z:-S1,0
4800 RDRAW @Z:2*S1,0
4810 RDRAW @Z:-S1,0
4820 RETURN
4830 RMOVE @Z:S1,S2
4840 RDRAW @Z:0,-2*S2
4850 RDRAW @Z:-2*S1,0
4860 RDRAW @Z:0,2*S2

```

```
4870 RDRAW @Z:2*S1,0
4880 RMOVE @Z:-S1,-S2
4890 RETURN
4900 RMOVE @Z:0,S2
4910 RDRAW @Z:S1,-S2
4920 RDRAW @Z:-S1,-S2
4930 RDRAW @Z:-S1,S2
4940 RDRAW @Z:S1,S2
4950 RMOVE @Z:0,-S2
4960 RETURN
4970 RMOVE @Z:S1/2,S2/8
4980 RDRAW @Z:-S1/4,S2/4
4990 RDRAW @Z:-S1/2,0
5000 RDRAW @Z:-S1/4,-S2/4
5010 RDRAW @Z:0,-S2/2
5020 RDRAW @Z:S1/4,-S2/4
5030 RDRAW @Z:S1/2,0
5040 RDRAW @Z:S1/4,S2/4
5050 RDRAW @Z:0,S2/2
5060 RMOVE @Z:-S1/2,-S2/8
5070 RETURN
5080 IF T(2)=1 THEN 5100
5090 RDRAW @Z:0,0
5100 RETURN
5110 T(16)=10^39
5120 T(17)=-T(22)
5130 FOR U1=1 TO T(1)
5140 T(16)=T(16) MIN Y(U1)
5150 T(17)=T(17) MAX Y(U1)
5160 NEXT U1
5170 T9=T(17)-T(16)
5180 T(17)=T9*(T9<>0)+(T9=0)
5190 RETURN
5200 T(6)=10^39
5210 T(7)=-T(12)
5220 FOR U1=1 TO T(1)
5230 T(6)=T(6) MIN X(U1)
5240 T(7)=T(7) MAX X(U1)
5250 NEXT U1
5260 T9=T(7)-T(6)
5270 T(7)=T9*(T9<>0)+(T9=0)
5280 RETURN
5290 DIM S0(5)
5300 S0(3)=INT(T(U9+1)/15)
5310 S0(3)=T(U9+3)/S0(3)
5320 S0(4)=INT(ABS(LGT(S0(3))))*SGN(LGT(S0(3)))
5330 S0(4)=INT(10^S0(4)+1.0E-4)
5340 IF S0(4)>0 THEN 5380
5350 S0(3)=1
5360 S0(4)=1
```

```
5370 GO TO 5460
5380 S0(5)=S0(3)/S0(4)
5390 S0(3)=S0(4)*10
5400 IF S0(5)>10 THEN 5420
5410 S0(3)=S0(4)*5
5420 IF S0(5)>5 THEN 5440
5430 S0(3)=S0(4)*2
5440 IF S0(5)>2 THEN 5460
5450 S0(3)=S0(4)
5460 S0(5)=T(U9+2)/S0(3)
5470 IF S0(5)>0 THEN 5490
5480 S0(5)=S0(5)-0.9999
5490 S0(1)=S0(3)*(INT(ABS(S0(5)))*SGN(S0(5)))
5500 S0(5)=(T(U9+3)+T(U9+2))/S0(3)
5510 IF S0(5)<0 THEN 5530
5520 S0(5)=S0(5)+0.9999
5530 S0(2)=S0(3)*(INT(ABS(S0(5)))*SGN(S0(5)))
5540 S0(5)=S0(2)-S0(1)
5550 S0(4)=INT(INT(ABS(S0(5)))*SGN(S0(5))/S0(3))
5560 IF S0(5)>1 THEN 5590
5570 S0(3)=S0(5)
5580 S0(4)=1
5590 IF S0(3)>1 THEN 5620
5600 S0(3)=S0(5)
5610 S0(4)=1
5620 T(U9+2)=S0(1)
5630 T(U9+3)=S0(2)-S0(1)
5640 T(U9+4)=S0(3)
5650 T(U9+5)=S0(4)
5660 T(U9+6)=1.75*T(U9+3)/T(U9+1)
5670 RETURN
```

Appendix G -- Text Plot Program Listing

```
4 GO TO 100
8 GO TO 400
12 GO TO 190
16 GO TO 410
20 GO TO 420
24 GO TO 550
28 GO TO 860
32 GO TO 1250
36 GO TO 1290
40 GO TO 760
100 SET KEY
101 L4=0
102 L3=0
105 REM BEGIN DATA ENTRY ****
110 DIM Z$(10000),B$(1)
120 S5=0
121 C5=1.79
122 C1=1.79
124 C2=2.82
125 C6=2.82
126 C3=1.79
128 C4=2.82
130 J$=CHR(10)
135 H$=CHR(8)
140 K$=CHR(11)
150 R$=CHR(13)
160 G$=CHR(7)
170 Z$=" "
180 T=1
185 REM DATA RETRIEVAL ****
190 PRINT "DO YOU WANT TO ENTER DATA FROM TAPE? ";
195 S5=0
200 INPUT A$
210 IF A$="N" THEN 330
220 PRINT "INSERT DATA TAPE AND ENTER FILE # ";
230 INPUT F
240 FIND F
250 READ @33:B
260 FOR I=1 TO B-1
270 READ @33:B$
280 Z$=Z$&B$
290 NEXT I
300 T=T+B-1
305 PRINT "DATA RETRIEVAL COMPLETED";G$
310 CLOSE
320 GO TO 400
330 PRINT "START ENTERING DATA FROM KEYBOARD"
335 PRINT "TO STOP PRESS UDK #2";J$
```

```
340 INPUT A$  
350 Z$=Z$&A$  
360 Z$=Z$&"#"  
370 Z=LEN(A$)+1  
380 T=T+Z  
390 GO TO 340  
400 END  
410 P=0  
420 P1=0  
430 IF P=T THEN 530  
440 P=P+1  
450 B$=SEG(Z$,P,1)  
460 PRINT B$;  
470 IF B$<>"#" THEN 510  
480 IF P1=1 THEN 530  
490 P1=1  
500 GO TO 430  
510 P1=0  
520 GO TO 430  
530 PRINT " "  
540 END  
545 REM EDITOR *****  
550 PRINT "ENTER STRING OF TEXT TO BE EDITED"  
560 INPUT A$  
570 E=LEN(A$)  
580 E2=POS(Z$,A$,1)  
590 IF E2<>0 THEN 700  
600 PRINT "STRING NOT FOUND"  
610 END  
620 PRINT "ENTER REPLACEMENT STRING"  
630 INPUT A$  
640 E3=LEN(A$)  
650 Z$=REP(A$,E2,E)  
660 T=T-E+E3  
670 END  
685 REM SAVE DATA *****  
690 PRINT "INSERT DATA TAPE AND ENTER FILE # ";  
700 INPUT F  
710 FIND F  
720 WRITE @33:T  
730 FOR I=2 TO T  
740 B$=SEG(Z$,I,1)  
750 WRITE @33:B$  
760 NEXT I  
770 CLOSE  
780 END  
795 REM SET CHARACTER SIZE *****  
800 I=1  
810 S1=0  
820 S2=0
```

```
890 S3=0
900 S4=0
910 S5=1
920 I=I+1
930 B$=SEG(Z$,I,1)
935 IF B$="%" THEN 1080
940 IF B$<>"^" AND B$<>"@" THEN 970
950 S5=2
960 GO TO 1080
970 IF B$<>"#" THEN 1060
975 IF I=T THEN 990
980 IF S6=0 THEN 1010
985 S3=S3-1
990 S4=S4 MAX S3+1
1000 S3=0
1010 S2=S2 MAX S1
1020 S1=0
1030 S3=S3+1
1040 S6=1
1050 GO TO 1080
1060 S6=0
1070 S1=S1+1
1080 IF I=T THEN 1100
1090 GO TO 920
1100 S2=130/S2
1110 S4=100/(S4*S5+S5-1)
1120 PRINT "LARGEST RECOMMENDED CHARACTER SIZE IS:";
1130 PRINT S2;" FOR WIDTH, AND ";S4;" FOR HEIGHT."
1140 IF S5=1 THEN 1190
1150 PRINT "YOU MAY USE LARGER CHARACTER SIZE THAN THE RECOMMENDED"
1160 PRINT "MAXIMUM FOR MAIN TEXT, IF YOU USE SMALLER CHARACTER SIZE"
1170 PRINT "FOR SUB/SUPER SCRIPTS."
1180 PRINT "ENTER CHARACTER SIZE FOR MAIN TEXT. ";G$
1200 INPUT C1,C2
1202 C5=C1
1204 C6=C2
1210 IF S5=1 THEN 1240
1220 PRINT "ENTER CHARACTER SIZE FOR SUPER/SUB SCRIPTS. ";G$
1230 INPUT C3,C4
1240 RETURN
1250 REM PLOTTING ROUTINE ****
1251 IF L4=1 THEN 1280
1252 DIM X(600),Y(600),W(50)
1253 SET NOCASE
1254 Z3=0
1256 L3=0
1258 RESTORE
1262 FOR I1=1 TO 548
1264 READ X(I1),Y(I1)
1265 NEXT I1
```

```
1266 FOR I1=1 TO 42
1268 READ W(I1)
1270 NEXT I1
1275 L4=1
1280 L=1
1290 L1=0
1300 L2=0
1302 PRINT "ENTER OUTPUT DEVICE #. (1=PLOTTER, 32=SCREEN) ... "
1305 INPUT O
1306 IF O=32 THEN 1320
1307 IF S5C>0 THEN 1310
1308 GOSUB 860
1310 PRINT @0,17:C1,C2
1320 PRINT @0:J$
1322 IF S5=2 THEN 1330
1324 PRINT @0:K$:K$
1330 IF L=T THEN 1740
1340 L=L+1
1350 B$=SEG(Z$,L,1)
1355 IF L3=1 THEN 2000
1360 IF B$<>"^" THEN 1390
1370 GOSUB 1480
1380 GO TO 1330
1390 IF B$<>"@" THEN 1420
1400 GOSUB 1570
1410 GO TO 1330
1420 IF B$<>"#" THEN 1441
1430 GOSUB 1680
1440 GO TO 1330
1441 IF B$<>"&" THEN 1450
1442 L3=1
1443 GO TO 1330
1450 PRINT @0:B$;
1460 L2=0
1470 GO TO 1330
1480 IF L1=1 THEN 1530
1490 PRINT @0,17:C1/6,C2/1.5
1495 PRINT @0:H$,K$;
1500 PRINT @0,17:C3,C4
1502 C5=C3
1504 C6=C4
1510 L1=1
1520 RETURN
1530 PRINT @0,17:C1/6,C2/1.5
1535 PRINT @0: " ";J$;
1540 PRINT @0,17:C1,C2
1542 C5=C1
1544 C6=C2
1550 L1=0
1560 RETURN
```

```
1570 IF L1=1 THEN 1630
1580 PRINT @0,17:C1/6,C4*0.67
1590 PRINT @0:H$;J$;
1600 PRINT @0,17:C3,C4
1602 C5=C3
1604 C6=C4
1610 L1=1
1620 RETURN
1630 PRINT @0,17:C1/6,C4*0.67
1640 PRINT @0:" ";K$;
1650 PRINT @0,17:C1,C2
1652 C5=C1
1654 C6=C2
1660 L1=0
1670 RETURN
1680 IF L2=1 THEN 1740
1685 PRINT @0,17:C3,C4
1690 IF S5=1 OR S5=0 THEN 1705
1700 PRINT @0:R$;
1705 PRINT @0,17:C1,C2
1710 PRINT @0:R$;
1715 C5=C1
1716 C6=C2
1720 L2=1
1730 RETURN
1740 END
2000 SET NOCASE
2005 GIN @0:Z1,Z2
2010 IF B$<>"a" THEN 2030
2020 Z3=1
2030 IF B$<>"b" THEN 2050
2040 Z3=2
2050 IF B$<>"c" THEN 2070
2060 Z3=3
2070 IF B$<>"d" THEN 2090
2080 Z3=4
2090 IF B$<>"e" THEN 2110
2100 Z3=5
2110 IF B$<>"g" THEN 2130
2120 Z3=6
2130 IF B$<>"h" THEN 2150
2140 Z3=7
2150 IF B$<>"j" THEN 2170
2160 Z3=8
2170 IF B$<>"i" THEN 2190
2180 Z3=9
2190 IF B$<>"k" THEN 2210
2200 Z3=10
2210 IF B$<>"l" THEN 2230
2220 Z3=11
```

2230 IF B\$<>"m" THEN 2250
2240 Z3=12
2250 IF B\$<>"n" THEN 2270
2260 Z3=13
2270 IF B\$<>"q" THEN 2290
2280 Z3=14
2290 IF B\$<>"p" THEN 2310
2300 Z3=15
2310 IF B\$<>"r" THEN 2330
2320 Z3=16
2330 IF B\$<>"s" THEN 2350
2340 Z3=17
2350 IF B\$<>"t" THEN 2370
2360 Z3=18
2370 IF B\$<>"u" THEN 2390
2380 Z3=19
2390 IF B\$<>"f" THEN 2410
2400 Z3=20
2410 IF B\$<>"x" THEN 2430
2420 Z3=21
2430 IF B\$<>"y" THEN 2450
2440 Z3=22
2450 IF B\$<>"w" THEN 2470
2460 Z3=23
2470 IF B\$<>"C" THEN 2490
2480 Z3=24
2490 IF B\$<>"D" THEN 2510
2500 Z3=25
2510 IF B\$<>"J" THEN 2530
2520 Z3=26
2530 IF B\$<>"L" THEN 2550
2540 Z3=27
2550 IF B\$<>"Q" THEN 2570
2560 Z3=28
2570 IF B\$<>"P" THEN 2590
2580 Z3=29
2590 IF B\$<>"S" THEN 2610
2600 Z3=30
2610 IF B\$<>"U" THEN 2630
2620 Z3=31
2630 IF B\$<>"F" THEN 2650
2640 Z3=32
2650 IF B\$<>"Y" THEN 2670
2660 Z3=33
2670 IF B\$<>"W" THEN 2690
2680 Z3=34
2690 IF B\$<>"X" THEN 2710
2700 Z3=35
2710 IF B\$<>"A" THEN 2730
2720 Z3=36

```

2730 IF B$<>"B" THEN 2750
2740 Z3=37
2750 IF B$<>"E" THEN 2770
2760 Z3=38
2770 IF B$<>"H" THEN 2782
2780 Z3=39
2782 IF B$<>"I" THEN 2786
2785 Z3=40
2786 IF B$<>"K" THEN 2790
2787 Z3=41
2790 IF Z3<>0 THEN 4002
3998 Z3=0
3999 L3=0
4000 GO TO 1450
4002 I5=0
4005 IF X(W(Z3)+I5)<0 THEN 4010
4007 DRAW @0:X(W(Z3)+I5)*C5+Z1,Y(W(Z3)+I5)*C6+Z2
4009 GO TO 4012
4010 MOVE @0:-X(W(Z3)+I5)*C5+Z1,Y(W(Z3)+I5)*C6+Z2
4012 IF I5+1=W(Z3+1)-W(Z3) THEN 4100
4015 I5=I5+1
4018 GO TO 4005
4100 L3=0
4101 SET CASE
4102 MOVE @0:C5+Z1,Z2
4103 Z3=0
4105 GO TO 1460
4110 DATA -0.6,0.3,0.37,0.05,0.28,0,0.17,0,0.08,0.05,0,0.11,0,0.18
4120 DATA 0.08,0.25,0.17,0.3,0.28,0.3,0.37,0.25,0.5,0.03,0.53,0,0.55
4130 DATA 0,0.57,0.03,0.6,0.05,-1.0E-3,-0.15,0.33,0.6,0.49,0.6,0.6
4140 DATA 0.51,0.6,0.43,0.49,0.34,0.33,0.3,0.49,0.27,0.6,0.19,0.6
4150 DATA 0.09,0.44,0.02,0.065,0,-1.0E-3,0.6,0.11,0.56,0.23,0.49,0.4
4160 DATA 0.18,0.4,0.09,0.31,0,0.26,0,0.2,0.04,0.17,0.11,0.17,0.14,0.6
4170 DATA 0.6,-0.6,0.41,0.54,0.56,0.43,0.6,0.31,0.56,0.31,0.49,0.34
4180 DATA 0.41,0.49,0.23,0.49,0.15,0.43,0.08,0.37,0.04,0.26,0,0.09
4190 DATA 0.04,0,0.11,0,0.23,0.09,0.26,0.37,0.3,-0.6,0.27,0.5,0.29,0.4
4200 DATA 0.3,0.2,0.29,0.1,0.27,0.05,0.25,0,0.21,0,0.19,0.05,0.17,0.1
4210 DATA 0.16,0.3,0.15,0.1,0.14,0.05,0.13,0,0.11,0,0.09,0.05,0.05,0.1
4220 DATA 0.03,0.2,0.01,0.4,0,0.5,0.01,0.6,0.03,-0.33,0.6,0.2,0.56
4230 DATA 0.17,0.52,0.2,0.45,0.33,0.45,0.53,0.52,0.53,0.56,0.47,0.56
4240 DATA 0.33,0.48,0.13,0.29,0.07,0.21,0,0.1,0.03,0.06,0.13,0.02,0.33
4250 DATA 0.06,0.47,0.1,0.57,0.1,0.6,0.06,0.6,0.04,0.47,0,-1.0E-3,0.17
4265 DATA 0.07,0.26,0.13,0.3,0.2,0.26,0.2,0.17,0.13,0,-0.2,0.17,0.27
4270 DATA 0.26,0.4,0.3,0.47,0.3,0.53,0.26,0.6,0.17,0.47,-0.15,-1.0E-3
4280 DATA 0.15,0.05,0.22,0.11,0.26,0.16,0.28,0.27,0.3,0.38,0.3,0.49
4290 DATA 0.28,0.55,0.26,0.6,0.22,0.6,0.16,0.55,0.08,0.49,0.04,0.43
4300 DATA 0.02,0.33,0,0.22,0,0.11,0.02,0.05,0.04,0,0.1,0,0.15,0.6,0.16
4310 DATA -0.4,0.3,0.3,0.06,0.3,0.04,0.35,0,0.4,0,0.45,0.04,-0.1,0.3,0
4320 DATA 0,-0.6,0.3,0.05,0.15,0.6,0,-1.0E-3,0,0.33,0.18,-0.06,0.3,0.13
4330 DATA 0.3,0.2,0.28,0.27,0.25,0.33,0.2,0.4,0.08,0.47,0.03,0.53,0

```

```

4340 DATA 0.6,0,-1.0E-3,-0.15,0.08,0.06,0.16,0.3,-0.1,0.12,0.11,0.08
4350 DATA 0.14,0.03,0.22,0,0.33,0,0.38,0.03,0.44,0.12,0.49,0.3,-0.4
4360 DATA 0.06,0.41,0.03,0.44,0,0.49,0,0.55,0.03,0.6,0.09,-1.0E-3,0.28
4370 DATA 0.11,0.3,0.22,0,0.38,0.08,0.49,0.15,0.55,0.2,0.6,0.3,-0.42
4380 DATA 0.6,0.32,0.56,0.3,0.54,0.34,0.51,0.42,0.51,0.46,0.52,0.48
4390 DATA 0.54,0.42,0.55,0.28,0.54,0.18,0.5,0.14,0.46,0.14,0.43,0.21
4400 DATA 0.39,0.28,0.37,0.35,0.36,0.42,0.36,0.44,0.37,0.44,0.38,0.42
4410 DATA 0.39,0.28,0.39,0.14,0.35,0,0.27,0,0.19,0.07,0.12,0.14,0.08
4420 DATA 0.28,0.04,0.39,0.04,0.49,0.08,0.56,0.08,0.6,0.06,0.58,0.04
4430 DATA 0.56,0.02,0.49,0,-1.0E-3,0.3,0.6,0.3,-0.18,0.3,0.12,0,-0.42
4440 DATA 0.3,0.36,0,-1.0E-3,-0.15,0.15,0.23,0.23,0.26,0.34,0.3,0.45
4450 DATA 0.3,0.53,0.26,0.6,0.19,0.56,0.08,0.53,0.04,0.45,0.01,0.38
4460 DATA 0,0.3,0,0.19,0.04,0.11,0.11,-0.6,0.3,0.24,0.3,0.12,0.27
4470 DATA 0.06,0.25,0,0.19,0,0.11,0.06,0.05,0.12,0.03,0.24,0,0.3,0
4480 DATA 0.42,0.03,0.48,0.05,0.54,0.11,0.54,0.19,0.48,0.25,0.42,0.27
4490 DATA 0.3,0.3,-1.0E-3,0.3,0.6,0.3,-0.3,0.3,0.24,0,-1.0E-3,0.25,0
4500 DATA 0.28,0.05,0.3,0.11,0.3,0.16,0.28,0.16,0.18,0.14,0.08,0.14
4510 DATA 0.03,0.19,0,0.27,0,0.38,0.03,0.44,0.05,0.49,0.1,0.6,0.25
4520 DATA 0.59,0.3,-0.36,0.3,0.24,0.3,0.12,0.27,0.06,0.23,0,0.17,0
4530 DATA 0.13,0.06,0.07,0.12,0.03,0.24,0,0.36,0,0.48,0.03,0.54,0.07
4540 DATA 0.6,0.13,0.6,0.17,0.54,0.23,0.48,0.27,0.36,0.3,-0.36,0.45
4550 DATA 0.24,-0.15,-0.6,0.3,0.53,0.24,0.47,0.21,0.27,0.15,0.13,0.09
4560 DATA 0,0,-1.0E-3,0.3,0.13,0.28,0.2,0.25,0.4,0.05,0.47,0.02,0.6
4570 DATA 0,-0.33,0.45,0.2,-0.15,-1.0E-3,0.3,0,0.13,0.07,0.04,0.13,0
4580 DATA 0.3,0,0.4,0.04,0.47,0.09,0.6,0.26,-0.15,0.3,0.1,0.26,0.05
4590 DATA 0.19,0,0.09,0,0.04,0.05,0,0.13,0,0.23,0.04,0.28,0.08,0.3
4600 DATA 0.15,0.33,0.08,0.38,0.04,0.48,0,0.55,0,0.6,0.04,0.6,0.09
4610 DATA 0.55,0.19,0.5,0.26,0.45,0.3,-1.0E-3,0.6,0.6,0.6,0.6,0.48
4620 DATA -0.16,0.6,0.16,0,-1.0E-3,0,0.33,0,-0.3,0.6,0,0,0.6,0,0.3
4630 DATA 0.6,-0.35,0.6,0.25,0.6,0.15,0.56,0.05,0.49,0,0.38,0,0.23
4640 DATA 0.05,0.11,0.15,0.04,0.25,0,0.35,0,0.45,0.04,0.55,0.11,0.6
4650 DATA 0.23,0.6,0.38,0.55,0.49,0.45,0.56,0.35,0.6,-0.15,0.56,0.15
4660 DATA 0.04,-0.45,0.56,0.45,0.04,-0.15,0.3,0.45,0.3,-0.04,0,0.3,0.6
4670 DATA 0.56,0,-1.0E-3,0,0.08,0,-0.52,0,0.6,0,-1.0E-3,0.6,0.6,0.6,-0.1
4680 DATA 0.3,0.5,0.3,-1.0E-3,0,0.6,0,-1.0E-3,0.6,0.6,-0.11,0.6,0.11
4690 DATA 0,-1.0E-3,0,0.22,0,-0.49
4692 DATA 0.6,0.49,0,-0.38,0,0.6,0,-0.6,0.49,0.6,0.6,0
4700 DATA 0.6,0.26,0.32,0,0,0.6,0,0.6,0.11,-1.0E-3,0.56,0.05,0.6,0.1
4710 DATA 0.6,0.3,0.19,0.5,0.56,0.55,0.6,0.6,0.56,-0.3,0.19,0.3,0,-0.2
4720 DATA 0,0.4,0,-0.4,0.48,0.2,0.48,0.1,0.44,0.05,0.4,0,0.32,0,0.28
4730 DATA 0.05,0.2,0.1,0.16,0.2,0.12,0.4,0.12,0.5,0.16,0.55,0.2,0.6
4740 DATA 0.28,0.6,0.32,0.55,0.4,0.5,0.44,0.4,0.48,-0.3,0.6,0.3,0,-0.1
4750 DATA 0.6,0.5,0.6,-0.1,0,0.5,0,-1.0E-3,0.44,0.05,0.44,0.05,0.32,0.1
4760 DATA 0.24,0.15,0.2,0.25,0.16,0.35,0.16,0.45,0.2,0.5,0.24,0.55
4770 DATA 0.32,0.55,0.36,0.55,0.44,0.6,0.44,-0.3,0.6,0.3,0,-0.15,0.6
4780 DATA 0.45,0.6,-0.15,0,0.45,0,-1.0E-3,0,0.14,0,0.05
4790 DATA 0.17,0,0.3,0,0.39,0.05,0.47,0.14,0.56,0.23,0.6,0.37,0.6,0.46
4800 DATA 0.56,0.55,0.47,0.6,0.39,0.6,0.3,0.55,0.17,0.46,0,0.6,0
4810 DATA -1.0E-3,-0.05,0.05,-0.15,0.11,-0.2,0.16,-0.2,0.22,-0.15,0.27
4820 DATA -0.05,0.33,0.6,0.38,0.7,0.44,0.75,0.49,0.75,0.55,0.7,0.6,0.6

```

```
4830 DATA -1.0E-3,0.6,0.6,0.38,0,0.15,-1.0E-3,0,0.6,0,-0.6,0.6,0,0.38
4840 DATA 0.6,0.15,-1.0E-3,0,0.6,0,-1.0E-3,0.45,0.6,0.45,-1.0E-3,0.15,0.6
4850 DATA 0.15,-0.38,0.6,0.22,0,-1.0E-3,0.35,0.6,0.35,-0.3,0.6,0.3,0.1
4860 DATA -1.0E-3,0,0.6,0,-1.0E-3,0.15,0.01,0.2,0.03,0.24,0.06,0.27,0.1
4870 DATA 0.29,0.15,0.3,0.2,0.29,0.24,0.27,0.27,0.24,0.29,0.2,0.31,0.1
4880 DATA 0.33,0.06,0.36,0.03,0.4,0.01,0.45,0,0.5,0.01,0.54,0.03,0.57
4890 DATA 0.06,0.59,0.1,0.6,0.15,0.59,0.2,0.57,0.24,0.54,0.27,0.5,0.29
4900 DATA 0.45,0.3,0.4,0.29,0.36,0.27,0.33,0.24,0.31,0.2,0.29,0.1,0.27
4910 DATA 0.06,0.24,0.03,0.2,0.01,0.15,0,0.1,0.01,0.06,0.03,0.03,0.06
4920 DATA 0.01,0.1,0,0.15,-0.6,0.6,0.5,0.6,0.4,0.59,0.3,0.57,0.2,0.54
4930 DATA 0.1,0.5,0.05,0.45,0,0.39,0,0.21,0.05,0.15,0.1,0.1,0.2,0.06
4940 DATA 0.3,0.03,0.4,0.01,0.5,0,0.6,0,-0.6,0.3,0,0.3,1,17,29,40,56
4950 DATA 77,97,110,130,136,141,152,169,176,209,215,229,246,250,265
4960 DATA 284,296,306,325,332,336,359,366,372,382,389,400,423,442,458
4970 DATA 470,475,480,486,492,531,549
```


Appendix H -- Block Plot Program Listing

```
110 REM THIS PROGRAM PLOTS BLOCK LETTERS
120 REM AUTHOR: LYSANDER NG OF NMFS-CHARLESTON
130 REM DATE WRITTEN: 3/84
140 DIM Z$(4000),Z(50)
145 SET DEGREES
150 Z$=""
160 Y=0
170 T=0
180 S=0
190 PRINT "ENTER DATA"
200 PRINT "ENTER ^#^ TO STOP ENTRY"
210 INPUT A$
230 IF A$="#^" THEN 400
240 Z$=Z$&A$
250 Y=Y+1
260 Z(Y)=LEN(A$)
270 S=S+Z(Y)
280 T=Z(Y) MAX T
290 GO TO 210
400 PRINT " "
410 PRINT "ENTER THICKNESS OF CHARACTER LINES IN GDU"
420 INPUT W
430 A=(150-W)/T
440 B=(100-W)/Y
450 PRINT " "
460 PRINT "ENTER HEIGHT OF CHARACTERS, MAXIMUM=";B
470 INPUT H
480 PRINT @1,17:A,H
490 T=1
500 FOR I=1 TO Y
510 A$=SEG(Z$,T,Z(I))
530 F1=INT(18/W)
540 F=-12/W
550 F=F+F1
560 MOVE @1:(1+SIN(F))*W/2,100-B*I+(COS(F)-1)*W/2
570 PRINT @1:A$
580 IF F=>360-12/W THEN 590
585 GO TO 550
590 T=T+Z(I)
600 NEXT I
610 END
```